Федеральное государственное автономное образовательное учреждение
высшего образования «Московский физико-технический институт «МФТИ»
Физтех-школа прикладной математики и информатики
Кафедра интеллектуальных систем

На правах рукописи

Баймурзина Диляра Римовна

Нейросетевые модели и диалоговая система для ведения разговора на общие темы

Специальность 05.13.18 —
«Математическое моделирование, численные методы и комплексы программ»

Диссертация на соискание учёной степени
кандидата технических наук

Научный руководитель:
кандидат физико-математических наук
Бурцев Михаил Сергеевич

Москва — 2021

Оглавление

Стр.

Стр.

Введение

Создание диалоговой системы, способной быстро, связно и осмысленно вести диалог на общие темы является одной из фундаментальных проблем в области искусственного интеллекта (ИИ). Развитие разговорного ИИ началось с диалоговых систем, основанных на правилах и шаблонах [1]. Последние достижения в области обработки естественного языка, например, предварительное обучение языковых моделей [2—5], архитектуры на основе памяти, и новые наборы диалоговых данных [6—10], расширили возможности для решения многих сложных проблем, возникающих при понимании человека машиной. В результате современные диалоговые системы, такие как чат-боты XiaoIce [11] или боты-участники конкурса «Alexa Prize Socialbot Grand Challenge»[1], комбинируют в себе модели машинного и глубокого обучения с вручную написанными сценариями на основе шаблонов [12].

Большинство современных диалоговых систем и голосовых помощников имеют модульную архитектуру, включающую в себя модуль понимания естественного языка, набор разговорных навыков и диалоговый менеджер. Модуль понимания естественного языка обычно представляет из себя набор нейросетевых моделей для классификации текста, разметки (классификации элементов) последовательности и моделей извлечения информации из баз знаний. Таким образом, классификация является одной из важнейших задач, так как позволяет реализовать следующие функции: определение текущей темы диалога, распознавание намерений, анализ тональности, извлечение сущностей и определение их типов, выбор рекомендаций. Однако, классификация текстов, как и любые другие задачи понимания естественного языка, в контексте диалоговых систем имеет особенности, связанные со специфичностью области использования. В частности, в данной работе рассматривается влияние стилистики разговорной речи.

Отдельные разговорные навыки в современных диалоговых системах представляют из себя сценарные, ранжирующие или генеративные модели. Навыки на основе сценариев могут демонстрировать высокое качество диалога [13], однако такой подход имеет несколько важных недостатков, таких как

[1] https://developer.amazon.com/alexaprize/challenges/current-challenge/

сложность интеграции знаний о пользователе, понимания контекста и состояния диалога, ограниченность покрытия тем и ситуаций. Особенно заметны эти проблемы становятся при общении с проактивными пользователями, которые фактически берут на себя ведение диалога. Многие системы также до сих пор плохо справляются с демонстрацией здравого смысла в диалоге, что было показано в работах [14; 15]. В данном исследовании сделана попытка внедрить использование моделей предсказания здравого смысла в диалог.

Задачей диалогового менеджера является управление переключением между навыками, в частности, шаблонными навыками узких предметных областей и навыками диалога на общие темы. При этом ошибки выбора навыков являются наиболее важной проблемой, так как они часто приводят к изменению направления разговора в неподходящий момент. Текущие подходы к отслеживанию состояния диалога и управлению диалогом в основном являются реактивными и полагаются на результаты классификации намерений пользователя в последней реплике. Таким образом, диалоговому менеджеру не хватает высокоуровневого понимания целей пользователя в диалоге и взаимопонимания с ним. Опыт команд-участников конкурса «Alexa Prize Challenge» показывает, что даже поверхностное моделирование понимания пользователя путем внедрения шаблонных фраз, подтверждающих понимание реплики пользователя в ответах системы, значительно улучшает опыт пользователя [16; 17]. Поэтому разработка стратегий управления диалогом, учитывающих цели пользователей, представляет из себя многообещающее направление.

У автора, как у члена команды DREAM – участника конкурса «Alexa Prize Socialbot Grand Challenge» – была уникальная возможность проверить передовые исследовательские идеи в реальных условиях, в связи с чем была сформулирована следующая цель диссертационной работы.

Целью данной работы является разработка и исследование ключевых нейросетевых моделей, навыков и алгоритмов для ведения диалога на естественном языке и их интеграция в модульную диалоговую систему, способную поддерживать разговор на широкий спектр тем.

Для достижения поставленной цели необходимо было решить следующие **задачи**:

1. Исследовать влияние домена обучения векторных представлений слов, включая векторные представления языковых моделей, на качество решения задачи классификации текстов.

2. Обучить и опубликовать в открытом доступе модели оценки тональности и токсичности, адаптированные для разговорных данных.
3. Предложить и разработать сценарные разговорные навыки для диалоговой системы.
4. Разработать и опубликовать разговорные навыки, использующие нейросетевые модели предсказания здравого смысла.
5. Исследовать качество здравого смысла, демонстрируемого системой в диалогах, и корреляцию здравого смысла и автоматических метрик.
6. Предложить подход и разработать архитектуру диалогового менеджера для диалоговой системы открытого домена.
7. Предложить подход и разработать метод выбора финального ответа, позволяющий приоритизировать сценарные навыки и повысить качество выбора финального ответа.

Научная новизна:

1. Впервые было проведено исследование влияния домена векторных представлений языковых моделей на качество решения задачи классификации текстов на русском языке.
2. Обучены и опубликованы оригинальные нейросетевые модели оценки тональности и токсичности, адаптированные для разговорных данных на русском и английском языках.
3. Предложены и опубликованы оригинальные разговорные навыки, в основе которых лежат сценарии.
4. Предложены и опубликованы оригинальные разговорные навыки, интегрирующие модели предсказания здравого смысла.
5. Разработана новая схема разметки здравого смысла в диалоге.
6. Выполнено оригинальное исследование корреляции здравого смысла и автоматических метрик.
7. Разработан и опубликован оригинальный алгоритм выбора финального ответа, основанный на тегах и приоритизирующий сценарные разговорные навыки.

Практическая значимость заключается в следующем:

– Обученные в рамках работы векторные представления `fastText` для различных языковых стилей позволяют улучшить качество решения задач обработки естественного языка для соответствующего домена.

– Предложенные нейросетевые методы и векторные представления `fastText` были применены в конкурсе Kaggle «Toxic Comment Classification Challenge»[2] (18 место из 4539, золотая медаль).
– Все разработанные и обученные модели векторных представлений и классификаторов, включая модели оценки тональности и токсичности для диалогового домена, опубликованы в библиотеке DeepPavlov[3] и имеют тысячи скачиваний[4].
– Предложенная методология использования предобученных векторных представлений разговорного домена была применена к обучению всех классификаторов диалоговой системы DREAM в рамках конкурсов «Alexa Prize Challenge 3» и «Alexa Prize Challenge 4».
– Предложенные сценарные разговорные навыки и навыки, интегрирующие модели предсказания здравого смысла, а также алгоритмы выбора набора навыков и выбора финального ответа были применены в диалоговой системе DREAM в рамках конкурса «Alexa Prize Challenge 3», «Alexa Prize Challenge 4» и выложены в открытый доступ в рамках диалоговой системы DREAM[5].
– По результатам данной работы оформлены свидетельства о государственной регистрации программ для ЭВМ № 2021662460 «Программа выбора финального ответа из реплик-кандидатов», № 2021662601 «Программа разговорных навыков, интегрирующих модели предсказания аспектов здравого смысла в диалоге», № 2021664221 «Программа разговорного навыка для проведения диалога о кино», № 2021664168 «Среда для создания сценарных разговорных агентов».

Методология и методы исследования. В данной работе были применены:

– метод численного эксперимента для исследования задач классификации текстов;
– основы теории вероятностей;
– методы машинного обучения и теории глубокого обучения;

[2] https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge

[3] http://docs.deeppavlov.ai/en/master/features/models/classifiers.html, http://docs.deeppavlov.ai/en/master/features/pretrained_vectors.html

[4] например, векторные представления `fastText` для разговорного домена скачаны более 3 тысяч раз

[5] https://github.com/deepmipt/dream

– методы разработки на языках Python, Bash, включая разработку программного кода для библиотек с открытым исходным кодом DeepPavlov и DeepPavlov Agent.

Основные положения, выносимые на защиту:

1. Векторные представления `fastText` и языковых моделей ELMo и BERT соответствующего целевой задаче домена улучшают качество решения задачи классификации текстов для английского и русского языков.
2. Предложенные разговорные навыки, интегрирующие нейросетевые модели предсказания здравого смысла в диалог, демонстрируют более высокий уровень наличия явного здравого смысла, чем шаблонные навыки.
3. Для предложенной разметки уровней здравого смысла в диалоге, проявление явного здравого смысла и отсутствие здравого смысла могут быть оценены с помощью анализа тональности и токсичности реакции пользователя на реплики.
4. Предложенный алгоритм выбора финального ответа на основе тегов, приоритизирующий сценарные навыки, повышает качество выбора финальной реплики по сравнению с базовым алгоритмом, основанном на уверенности навыков, для модульной диалоговой системы открытого домена.

Достоверность полученных результатов обеспечивается экспериментами на наборах диалоговых данных, а также применением в соревнованиях Kaggle «Toxic Comment Classification Challenge», «Alexa Prize Challenge 3» и «Alexa Prize Challenge 4». Результаты находятся в качественном соответствии с результатами, полученными другими авторами.

Апробация работы. Результаты работы были представлены автором на следующих научных конференциях и семинарах:

– XXV Международная научная конференция студентов, аспирантов и молодых ученых «Ломоносов», доклад «Распознавание интентов с помощью нейросетей», Баймурзина Диляра, 9 - 13 апреля 2018, Москва;
– Конференция «Data Fest 5»[6], 28 апреля 2018, Москва;
– The 56th Annual Meeting of the Association for Computational Linguistics, Systems Demonstrations, демо-стенд «Deeppavlov: Open-source library for

[6] https://datafest.ru/5/

dialogue systems», Burtsev Mikhail, et al., 15 - 20 July 2018, Melbourne, Australia;

– XXV Международная конференция по компьютерной лингвистике и интеллектуальным технологиям «Диалог», доклад «Language model embeddings improve sentiment analysis», Baymurzina Dilyara, Kuznetsov Denis, Burtsev Mikhail, 29 мая - 1 июня 2019, Москва;
– Конференция «AI Journey», постер «Conversational BERT for English and Russian languages», Baymurzina Dilyara, Kuratov Yury, Pugachev Leonid, 8 – 9 ноября 2019, Москва;
– XXII Международная конференция по компьютерной лингвистике и интеллектуальным технологиям «Диалог», доклад «Evaluation of Conversational Skills for Commonsense», Baymurzina Dilyara, et al., 16 - 19 июня 2021, Москва.

Личный вклад. Результаты, представленные на конференции «Ломоносов» в докладе [18], получены автором самостоятельно. В работах [19] (индексируется Scopus), [20], «Conversational BERT for English and Russian languages» (постер на конференции «AI Journey») автором реализованы и обучены модели классификации текстов. В работе [21] и [22] (индексируется RSCI) автором была разработана часть аннотаторов, разговорных навыков, включая представленные в данной работе сценарные навыки и навыки, интегрирующие здравый смысл в диалог. В работе [23] (индексируется Scopus) автором разработаны разговорные навыки, интегрирующие здравый смысл в диалог, предложена схема разметки здравого смысла в диалоге, а также проведено исследование корреляции здравого смысла с автоматическими метриками. В работе [24] автором разработан алгоритм выбора финального ответа в диалоговой системе, а также разработана часть аннотаторов и разговорных навыков. Программы ЭВМ [25; 26] разработаны автором самостоятельно. В программе ЭВМ [27] автором разработана версия выборщика ответа на основе тегов. В программе ЭВМ [28] автор участвовала в доработке.

Публикации. Основные результаты по теме диссертации изложены в 7 печатных изданиях, 1 из которых издано в журналах, индексируемых RSCI, 2 — в периодических научных журналах, индексируемых Web of Science и Scopus, 2 — в тезисах докладов. Зарегистрированы 4 программы для ЭВМ.

Объем и структура работы. Работа состоит из введения, 5 глав, заключения. Полный объём работы составляет 136 страниц, включая 17 рисунков и 10 таблиц. Список литературы содержит 112 наименований.

Благодарности. Автор выражает искреннюю признательность своему научному руководителю, кандидату физико-математических наук, Бурцеву Михаилу Сергеевичу за помощь и наставничество в подготовке диссертации. Автор благодарит всех членов лаборатории нейронных систем и глубокого обучения МФТИ и участников команды DREAM за помощь в проведении исследований. Автор также выражает особую благодарность Кузнецову Денису, Игнатову Федору, Куратову Юрию, Юсупову Идрису и Корневу Даниле за переданный опыт и продуктивное сотрудничество. И конечно же, автор выражает глубокую признательность своей семье и Илье Жарикову за поддержку и участие.

Глава 1. Диалоговые системы

В разделе 1.1 дается краткое описание самых значимых моментов развития области диалоговых систем. В разделе 1.2 описываются основные выделяемые виды диалоговых систем. В разделе 1.3 представлены основные программные библиотеки для построения диалоговых систем.

1.1 История диалоговых систем

Люди задавались вопросом, смогут ли программируемые компьютеры стать умными и насколько задолго до того, как первый компьютер был создан. Сегодня искусственный интеллект (ИИ, англ.: Artificial Intelligence, AI) – это быстро развивающаяся область исследований, имеющая множество практических приложений. Искусственный интеллект, в основном, воспринимается как способ автоматизации рутинного человеческого труда, например, обработки речи или изображений, постановки диагноза в медицине и поддержки фундаментальных научных исследований.

В период раннего развития искусственного интеллекта в данной области быстро решались проблемы, которые интеллектуально сложны для людей, но относительно очевидны для компьютеров, такие проблемы, которые могут быть описаны в виде формальных математических правил. Сейчас же есть понимание, что *истинная задача искусственного интеллекта* заключается в решении задач, которые людям легко выполнять, но трудно для людей описать формально, то есть задач, которые человек решает интуитивно, такие как распознавание произносимых слов или лиц на картинках.

Нынешние исследования в области искусственного интеллекта посвящены решению именно таких интуитивных задач. Эти решения в том числе могут позволять компьютерам учиться на собственном опыте и понимать мир с точки зрения иерархии понятий. Предполагая получение знаний компьютерами из собственного опыта, этот подход позволяет избежать необходимости определения необходимых машине знаний человеком-создателем. Иерархия понятий позволяет компьютеру изучать сложные концепции, создавая их из более простых.

Если мы нарисуем граф, показывающий, как эти концепции построены друг над другом, то граф будет глубокий, с большим числом слоев. По этой причине этот подход называется глубоким обучением (англ.: Deep Learning, DL) искусственного интеллекта.

Одним из популярных направлений применения и развития искусственного интеллекта является создание *диалоговых систем* – специальных программ, предназначенных для общения с пользователем. Сама идея диалоговых систем, как концепции существования некоторого нечеловеческого существа или машины, способной общаться с человеком на привычном ему языке, возникла еще в древнейших цивилизациях. Античная мифология демонстрирует нам множество концепций разговаривающих нечеловеческих существ, например, нимфы и сатиры.

Первые диалоговые системы в виде компьютерных программ появились еще во второй половине XX века. Одной из самых известных первых диалоговых систем является ELIZA [1]. Так, в 1965 году произошло знаменательное событие: Джозеф Вайзенбаум (1923 – 2008) из Массачусетского технологического института представил ELIZA – интерактивную программу, которая ведет диалог на английском языке на любую тему. Чуть позже ELIZA была доработана и стала способна вести диалог, имитируя психотерапевта, что сделало ее очень популярной. ELIZA была разработана для моделирования клиент-центрированной психотерапии, которая основана на разделе клинической психологии и методы которой включают в себя установление контакта с пациентом за счет отражения психотерапевтом высказываний пациента. Клиент-центрированная психотерапия – это редкий тип разговора, в котором психотерапевт может «занять позицию отсутствия знаний о реальном мире». Если пациент говорит: «Я часто езжу на озеро.», а психотерапевт говорит: «Расскажите мне об озере.», он не предполагает, что пациент не знает, что такое озеро, а скорее предполагает, что пациент говорит об озере с некоторой целью. Такая позиция сама по себе облегчает мимикрию диалоговых систем под человека.

Принцип работы ELIZA заключается в выделении значимых слов во фразе пользователя и подстановке их в специальные шаблоны-ответы. Например, если пациент говорит: «You *love* me.», то ELIZA использует шаблон «You VERB me.», извлекает ключевое слово «love» и подставляет его в шаблон-ответ, возвращая реплику «WHAT MAKES YOU THINK I *LOVE* YOU?». Также ELIZA содержит набор правил для более универсального преобразования выражений из репли-

ки пользователя при подстановке в реплику системы, например, «my» («мой») в «your» («твой»).

Упомянутая выше мимикрия диалоговых систем под человека является одной из важнейших задач диалоговых систем на данный момент и до сих пор не решена. Однако стоит обратить внимание, что существует диалоговая система, прошедшая своеобразный тест Тьюринга еще в 1972 году. В 1971 году была представлена диалоговая система PARRY [29], использующая аналогичные ELIZA шаблонные реплики и систему, моделирующую собственное ментальное состояние. Например, некоторые темы могли вызывать у PARRY определенные эмоции, что проявлялось в использовании специальных наборов реплик, соответствующих вызванной эмоции. Создатели PARRY даже заложили в нем возможность делать вид, что диалоговая система испытывает галлюцинации, то есть перестает реагировать на реплики собеседника и делится своими мыслями, относящимися к «галлюцинации». В 1972 году PARRY прошел своеобразный тест Тьюринга – психиатры не смогли отличить текстовые транскрипции диалогов с PARRY от диалогов с настоящими больными шизофренией [30].

В 1977 году была представлена диалоговая система GUS [31] для решения задачи планирования путешествия. Фактически GUS является задаче-ориентированной (task-oriented) диалоговой системой (подробнее в следующем разделе 1.2), то есть диалог с GUS направлен на выполнение определенных задач, таких как бронирование авиабилетов. В диалоговой системе GUS был предложен фреймовый подход (англ.: frame-based approach). Фрейм – это некая структура знаний, представляющая информацию, которую система может извлечь из реплик пользователя, и состоит из набора слотов, каждый из которых может принимать значения из заданного набора. Этот набор слотов и определяет, какую информацию системе необходимо узнать у пользователя для выполнения задачи, например, даты вылета и прилета, город вылета и место назначения для задачи бронирования авиабилетов. Набор фреймов иногда называют *онтологией* предметной области.

С активным развитием нейросетевых моделей для решения задач обработки естественного языка, фреймовый подход стал широко использоваться в коммерческих диалоговых системах и лежит в основе большинства современных виртуальных ассистентов. Например, в 2011 году «Apple» сделала облачного персонального помощника «Siri» [32] неотъемлемой частью своего программного обеспечения. Данное приложение использует обработку есте-

ственной речи, чтобы отвечать на вопросы и давать рекомендации. «Siri» приспосабливается к каждому пользователю индивидуально, изучая его предпочтения в течение долгого времени. Причем «Siri» не просто выдает результаты на запросы пользователя, но и может пообщаться с ним на общие темы.

Эта технология стала первопроходцем в разработке многозадачных голосовых помощников. Следом за ней вышли приложения «Google Now» в 2012 году, «Microsoft Cortana» и «Amazon Echo» в 2014 году. «Microsoft Cortana» – виртуальный голосовой ассистент от «Microsoft». Персональный помощник «Cortana» призван предугадывать потребности пользователя. При желании, ей можно дать доступ к личным данным, таким как электронная почта, адресная книга, история поисков в сети и т. п. – все эти данные она будет использовать для предсказания потребительских нужд пользователя. «Amazon Alexa» – это виртуальный голосовой ассистент от «Amazon». В отличие от «Microsoft Cortana» и «Google Now», «Alexa» позволяет взаимодействовать с ней исключительно голосом, не предоставляя текстовый способ общения.

18 мая 2016 года компания «Google» объявила о выпуске голосового помощника «Google Assistant». В отличие от уже существующего сервиса «Google Now», «Google Assistant» может не только отвечать на простые запросы, но и распознавать вопросы на естественном языке. Также «Google Assistant» может отвечать на дополнительные вопросы в контексте уже предоставленного ответа. В 2017 году компания «Яндекс» представила голосового помощника «Алису». По заявлению создателей, «Алиса» не ограничивается набором заранее заданных ответных реплик, а также использует нейросетевые модели [33].

В 2020-х годах диалоговые системы становятся все более популярными, так как возможности моделей обработки естественного языка значительно увеличиваются и позволяют закрывать все больше потребностей пользователей. Например, все крупные компании с большим потоком клиентов активно используют ИИ для обеспечения базовой клиентской поддержки, в том числе голосовой. Виртуальные помощники выполняют все больше обязанностей, которые раньше выполняли ассистенты-люди: запись задач, составление календаря, бронирование и покупка различных товаров и услуг, развлекательные беседы.

1.2 Виды диалоговых систем

С точки зрения содержания диалога, выделяют две основные классификации диалоговых систем:

1. **назначение**:
   - *общего назначения* (англ.: general, chat-bot, socialbot) – системы, предназначенные для обычного разговора, без специальной задачи (чат-боты, социальные боты),
   - *задаче-ориентированные* (англ.: task-oriented) – системы, диалог с которыми решает определенную задачу,
2. **доменная область**:
   - *открытого домена* (англ.: open domain) – диалоговые системы, способные говорить на любые темы,
   - *закрытого домена* (англ.: closed domain) – диалоговые системы, способные говорить на одну или несколько строго определенных тем.

Диалоговые системы, которые используются в службе поддержке пользователей и представляют из себя роботов, ведущих диалог по специальному сценарию, способных ответить на ограниченных ряд наиболее популярных вопросов и даже выполнить некоторые простые операции по бронированию, составлению заявок, покупкам, являются *задаче-ориентированными* диалоговыми системами *закрытого домена*. Описанная в разделе 1.1 ELIZA является диалоговой системой *общего назначения закрытого домена*, так как она способна говорить в рамках темы психоанализа, однако при этом она не выполняет какой-то определенной задачи. Диалоговые системы, участвовавшие в конкурсах ConvAI [34; 35], представляют из себя *задаче-ориентированные* диалоговые системы *открытого домена*, задачей которых являлось провести диалог с пользователем на заданную тему. Если рассматривать искусственный интеллект как диалоговую систему, то в данной классификации она будет относиться к диалоговым системам *общего назначения открытого домена*. В эту же категорию попадают диалоговые системы, участвующие в конкурсе «Alexa Prize Challenge», университетском соревновании чат-ботов, поддерживающих разговор на общие темы с пользователями колонок «Amazon Alexa». Диалоговая

система «Replika»[1] также является системой открытого домена, ее основная задача – стать компаньоном, который поддерживает и понимает пользователя, то есть основной фокус в чат-боте сделан на установлении эмоциональной связи с пользователем. На конец октября 2020 года ежемесячно приложением «Replika» пользовалось около миллиона пользователей, что говорит о большой популярности диалоговых систем открытого домена.

Диалоговые системы, сочетающие в себе возможности разных видов (например, задаче-ориентированные системы, которые кроме того поддерживают диалог на свободные темы), называются *гибридными*. Яркими примерами гибридных диалоговых систем являются голосовые помощники, такие как «Яндекс Алиса», «Amazon Alexa», «Google Assistant», «Siri» от «Apple», «Cortana» и «Xiaoice» [11] от «Microsoft». Они не только исполняют некоторые задачи персонального ассистента, такие как установка напоминаний, помощь в подборе и бронировании товаров и услуг, но и могут поддержать обычные беседы на любые темы. В 2014 году команда «Microsoft» выпустила чат-бота «Xiaoice» [11], основной целью которого было также установление дружеских отношений с пользователем. По результатам исследования [11] от «Microsoft» большинство пользователей догадывались, что общаются с ботом, а не с человеком только спустя 10 минут после начала беседы. Успех «Xiaoice» привел к его развитию в полноценного персонального ассистента, обладающего сотнями различных навыков, и платформу для создания чат-ботов. Кроме того, «Xiaoice» умеет писать стихи и петь, рисовать, работать с финансовой отчетностью. «Xiaoice» умеет понимать не только естественную речь, но и анализировать изображения. «Xiaoice» также, как и «Replika», проявляет эмпатию к пользователю, показывает ему свою заботу и понимание. «Xiaoice» обладает не только встроенными сценариями, но и может вести себя совершенно непредсказуемо. В основе «Xiaoice» заложено постоянное дообучение за счет общения с реальными пользователями, что, например, привело к проблемам при запуске англоязычного аналога на американском рынке.

Другой важной характеристикой диалоговых систем является их **архитектура**. На верхнем уровне архитектуры диалоговых систем делятся на *цельные (end-to-end)* и *модульные (module-based)*. Цельные диалоговые системы состоят из одной модели, которая получает на вход текст реплики пользователя и выдает финальный ответ. Например, нейросетевая модель,

[1] https://replika.ai

принимающая на вход последовательность символов или токенов реплики пользователя и выдающая последовательность символов или токенов ответа системы (sequence-to-sequence), является цельной диалоговой системой. Однако такие модели имеют ряд значительных недостатков: невозможность получить интерпретируемое внутреннее представление, невозможность контролировать дальнейшее развитие диалога, необходимость дообучения системы в случае расширения домена или задач системы, а также огромное число параметров для достаточно сложной системы и лексическая ограниченность для более простых систем. Именно поэтому, предложенные еще в 1970-х годах, модульные архитектуры легли в основу современных диалоговых систем.

Пример модульной задаче-ориентированной диалоговой системы [36] представлен на Рисунке 1.1. В зависимости от способа передачи информации от пользователя диалоговой системе, в архитектуру входят модули распознавания и генерации речи для преобразования звукового сигнала в текст и обратно. *Модуль распознавания речи* (англ.: Automatic Speech Recognition, ASR) обычно представляет из себя обученную на большом объеме данных нейросетевую модель, принимающую на вход звуковой сигнал и выдающую список слов-гипотез с соответствующими показателями уверенности (стандартно, от 0 до 1). *Модуль генерации речи* (англ.: text-to-speech, TTS) обычно использует конканекативный подход (англ.: concatenative TTS) [37], заключающийся в соединении фрагментов фраз из заранее записанной большой базы данных коротких фрагментов речи одного человека. Такой подход позволяет легко преобразовывать звуковой сигнал, например, для придания эмоциональной окраски. Однако в последние годы появились нейросетевые модели [38], справляющиеся с переводом текста в естественную речь значительно лучше.

Основу задаче-ориентированной диалоговой системы, представленной на Рисунке 1.1, составляют три части: модуль понимания естественного языка (англ.: Natural Language Understanding, NLU), диалоговый менеджер (англ.: Dialogue Manager, DM) и модуль генерации естественной речи (англ.: Natural Language Generation, NLG). *Модуль понимания естественного языка* предназначен для извлечения различной информации из реплики пользователя, например, исправления опечаток (англ.: spell-checking), определения тональности (англ.: sentiment analysis) и домена (англ.: domain detection), распознавания намерений (англ.: intent recognition), классификации темы (англ.: topic classification), извлечения именованных сущностей (Named Entity

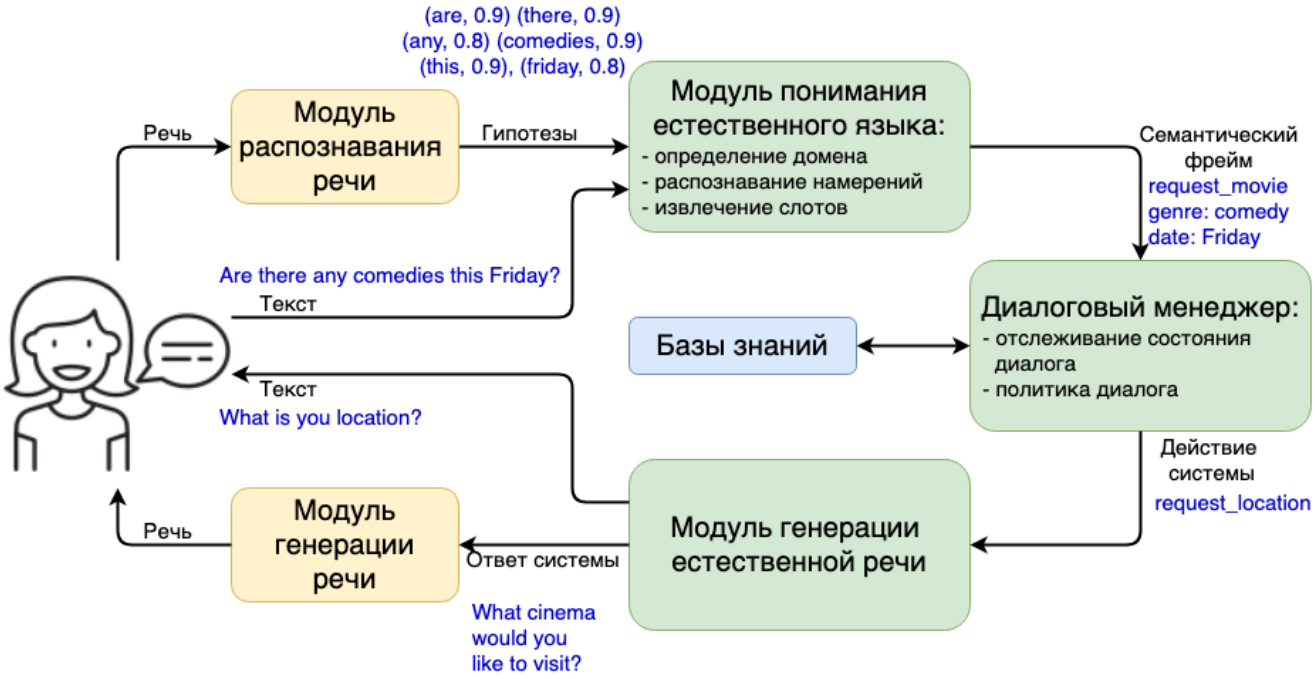

Рисунок 1.1 — Пример модульной диалоговой системы – задаче-ориентированная система для бронирования билетов в кинотеатр. Диалоговая система представлена в [36].

Recognition, NER) и прочего. Структурированная извлеченная информация представляет из себя семантический фрейм, который легко может быть обработан диалоговым менеджером, что является ключевым моментом для задаче-ориентированных систем. Диалоговый менеджер принимает решение о следующем действии системы на основе состояния диалога, включающего историю диалога и политику (стратегию, поставленную задачу) ведения диалога. Диалоговый менеджер, изображенный на Рисунке 1.1, может быть представлен нейросетевой моделью, предсказывающей одно из возможных действий системы. Принятое решение передается в модуль генерации естественной речи, которая, в случае задаче-ориентированных систем, обычно представляет из себя модель на основе эвристик, выбирающую шаблон, соответствующий действию системы, и заполняет в нем слоты в случае необходимости. Итоговый ответ системы передается пользователю.

1.3 Библиотеки для построения диалоговых систем

На данный момент наиболее распространены модульные диалоговые системы. В данном разделе описаны библиотеки для обучения и применения моделей обработки естественного языка, которые могут быть использованы как составляющие диалоговых систем, а также представлены одни из самых популярных библиотек для сборки архитектуры модульных диалоговых систем.

Неотъемлемой частью диалоговых систем являются модели **обработки естественного языка**, самыми популярными библиотеками для которых являются NLTK[2] [39], spaCy[3], fastText[4] [40], Gensim[5] [41], CoreNLP[6] [42], TextBlob[7], и для нейросетевых моделей – TensorFlow[8] [43], Keras[9] [44], PyTorch[10] [45], HuggingFace[11] [46], DeepPavlov[12] [19]. Самые популярные поддерживают огромное количество языков: NLTK обеспечивает легко используемый интерфейс для более 50 корпусов и текстовых ресурсов, spaCy поддерживает специфичные пайплайны для обработки текста на 18 языках, векторные представления fastText представлены для более, чем 150 языков. Фреймворки TensorFlow, Keras и PyTorch представлены для сборки, обучения и использования нейросетевых моделей. Библиотеки HuggingFace и DeepPavlov на языке Python предназначены для обучения и использования различных моделей обработки естественного языка. Автор является одним из разработчиков библиотеки DeepPavlov, имплементировала и обучила модели классификации текстов [19], в том числе представленные в Главе 2.

С точки зрения **сборки архитектуры диалоговой системы**, активно используются библиотеки Rasa[13] [47] и DeepPavlov Agent[14]. Основное отличие

[2] https://www.nltk.org
[3] https://spacy.io
[4] https://fasttext.cc
[5] https://pypi.org/project/gensim/
[6] https://corenlp.run/
[7] https://textblob.readthedocs.io/en/dev/
[8] https://www.tensorflow.org
[9] https://keras.io/
[10] https://pytorch.org/
[11] https://huggingface.co/
[12] https://deeppavlov.ai/
[13] https://rasa.com/
[14] https://deeppavlov.ai/agent, https://github.com/deepmipt/dp-agent

библиотек состоит в том, что Rasa специализируется на создании задаче-ориентированных систем для индустрии, кастомизация в которых происходит за счет создания и интеграции уникальных задаче-ориентированных сценариев. DeepPavlov Agent в свою очередь поддерживает создание диалоговых систем разного вида и по назначению, и по доменной области.

1.3.1 RASA

Фреймворк RASA предназначен для упрощения построения задаче-ориентированных диалоговых систем для индустрии. Основу фреймворка RASA составляют RASA NLU и RASA Core. RASA NLU – это фреймворк для понимания естественного языка, то есть распознавания намерений пользователя и извлечения сущностей из его реплики. Эта часть фреймворка предназначена для извлечения полезной информации и представления реплики пользователя в более удобном для понимания системой виде. RASA Core – это фреймворк, отвечающий за построение пайплайна диалоговой системы. Диалоговый менеджмент основан на моделях машинного обучения, которые определяют следующее действие системы на базе выхода NLU компонент, диалогового контекста и обучающих данных.

Основными моделями понимания естественного языка являются модели извлечения слотов (сущностей) и распознавание намерений пользователя. Для этого создаются обучающие выборки с примерами, размеченными классами намерения и выделенными сущностями. На Рисунке 1.2 приведен пример[15] тренировочного файла с разметкой намерений и сущностей. Для обучения модели также необходимо создавать конфигурационный файл с выбором и параметрами модели.

Тренировочные данные для самой диалоговой системы RASA Core называются историями (англ.: stories). История – это часть диалога между пользователем и системой. Входные данные от пользователя представлены в удобном для системы формате – сущности и намерения из реплики пользователя, а ответы системы представляют из себя действия (англ.: actions). Каждое действие системы представляет из себя некоторую метку, которая определяет,

[15] https://akhilck.medium.com/building-a-bot-using-rasa-nlu-core-5d8c90d254bc

какой именно шаблон будет использоваться для составления ответа. В ответах системы также могут быть использованы слоты (англ.: slots) – специальные переменные, в которые при формировании реплики могут быть подставлены некоторые текстовые значения. На Рисунке 1.3 приведен пример[15] истории для диалога, состоящего из приветствия и выдачи шутки.

Ответы выбираются среди заданных шаблонов для выбранного действия системы, слоты заполняются в соответствии с извлеченной информацией из реплики пользователя. На Рисунке 1.4 приведен пример[15] шаблонов для действий системы, отвечающих за приветствие в диалоге.

intent:name
– My name is [Juste]*(name)*
– I am [Josh]*(name)*
– I'm [Lucy]*(name)*
– People call me [Greg]*(name)*
– It's [David]*(name)*
– My name is [John]*(name)*

Рисунок 1.2 — Пример тренировочного файла с разметкой намерений и сущностей для получения модуля RASA NLU.

story_joke_02
∗ greet
– utter_name
∗ name{«name»:«Lucy»}
– utter_greet
∗ joke
– action_joke

Рисунок 1.3 — Пример истории из модуля RASA Core.

templates:
utter_name:
– text: «Hey there! Tell me your name.»
utter_greet:
– text: «Nice to you meet you name. How can I help?»

Рисунок 1.4 — Пример шаблонов, в том числе с использование слота, из модуля RASA Core.

Из структуры работы RASA Core становится ясно, что основной фокус RASA приходится на задаче-ориентированные диалоговые системы. Однако в новой версии RASA 2.0 было представлено также решение для диалогов открытого домена, чтобы бот мог на минимальном уровне поддерживать диалог вне задаче-ориентированного сценария, так как невозможность ответить на элементарные вопросы вне сценария плохо влияла на впечатления пользователя.

1.3.2 DeepPavlov Agent

DeepPavlov Agent[16] – это фреймворк с открытым исходным кодом для разработки масштабируемых и готовых к работе многофункциональных вир-

[16] https://deeppavlov.ai/agent, https://github.com/deepmipt/dp-agent

туальных помощников, сложных диалоговых систем и чат-ботов. За счет использования контейнеров в диалоговые системы на основе DeepPavlov Agent легко встраиваются не только модели из библиотеки DeepPavlov, но и любые другие модели для понимания и генерации естественного языка, в том числе другие диалоговые системы целиком.

Верхнеуровневая архитектура диалоговой системы в терминах фреймворка DeepPavlov Agent представлена на Рисунке 1.5. *Состояние диалога* (англ.: `Dialogue State`) – это структурированная информация, содержащая в себе историю диалога, включая реплики-кандидаты, аннотации всех реплик и реплик-кандидатов, а также специальные атрибуты пользователя и системы. *Аннотаторы* (англ.: `Annotators`), *аннотаторы реплик-кандидатов* (англ.: `Candidate Annotators`), *аннотаторы реплики системы* (англ.: `Response Annotators`) представляют из себя набор моделей понимания естественного языка, которые получают на вход текст реплики и состояние диалога, обычно включают в себя исправление опечаток, различные виды классификации текста и токенов, извлечение сущностей, а также другие модели анализа текста. *Выборщик навыков* (англ.: `Skill Selector`) формирует список навыков, которые будут вызваны для генерации реплик-кандидатов. *Навыки* (англ.: `Skills`) могут быть разных видов: шаблонные, ранжирующие, генеративные (подробнее в Разделе 1.4). *Выборщик ответа* (англ.: `Response Selector`) использует состояние диалога, реплики-кандидаты и их аннотации для финального решения. Все компоненты имеют доступ к состоянию диалога, все кроме `Skill Selector` и `Response Selector` могут его менять. Все процессы выполняются в асинхронном режиме, точками синхронизации являются `Skill Selector` и `Response Selector`. Агент представляет из себя отдельное приложение-контейнер, который делает запросы к базе диалогов, делает запросы к различным компонентам пайплайна, собирает ответы этих компонент и записывает их в структурированное состояние диалога, которое также записывается в базу диалогов.

Фреймворк DeepPavlov Agent использует технологию контейнеров для поднятия компонент диалоговой системы. Контейнеры позволяют упаковать само приложение и фиксированную среду запуска в единую систему, в связи с чем, во-первых, решается проблема конфликтов между зависимостями различных компонент, во-вторых, контейнеры легко версионируются, в-третьих, внутри контейнера компонента может быть реализована любыми инструмен-

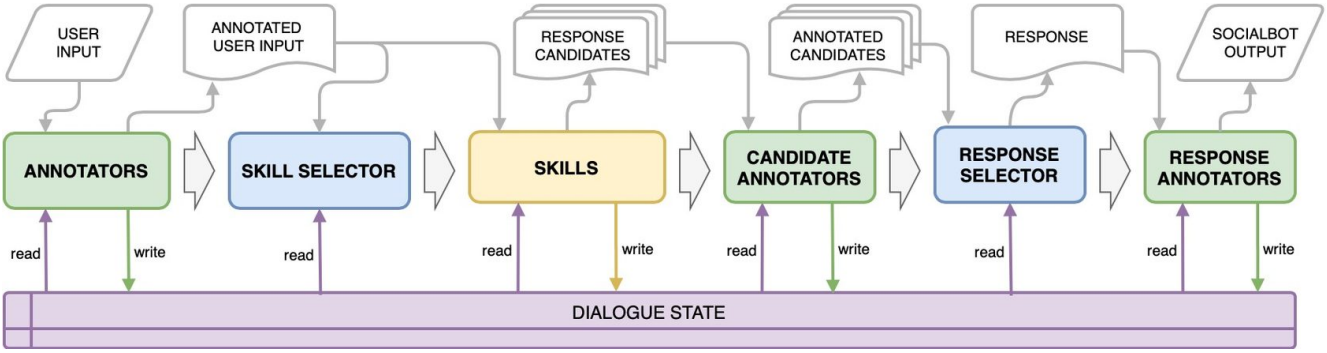

Рисунок 1.5 — Верхнеуровневая архитектура диалоговых систем во фреймворке DeepPavlov Agent.

тами. Контейнеры позволяют скомбинировать в единую диалоговую систему компоненты различного происхождения – базы данных, нейросетевые модели, модели машинного обучения, модели на основе различных эвристик. Также использование контейнеров помогает распределять и ограничивать ресурсы для каждой компоненты, что важно в условиях использования на реальных пользователях и онлайн работы. Со стороны оркестрирующего приложения всегда есть возможность проверять состояние и доступность всех контейнеров в системе.

Управление использованием компонент-контейнеров происходит с помощью контейнера агента DeepPavlov Agent, который, используя информацию из базы данных, рассылает необходимые запросы в сервисы, агрегирует полученные данные, сохраняет структурированную информацию в базу данных, а также синхронизирует работу различных сервисов.

1.4 Разговорные навыки в диалоговых системах

Начиная с данного раздела, будем использовать терминологию библиотеки DeepPavlov Agent. Все модели генерации ответных реплик будем называть разговорными *навыками* (англ.: conversational *skills*). В Разделе 1.4.1 описаны основные виды разговорных навыков. В Разделе 1.4.2 представлены наиболее распространенные инструменты для построения разговорных навыков.

1.4.1 Виды разговорных навыков

Разговорные навыки, как и диалоговые системы, могут быть классифицированы как на основе покрываемой области диалога, так и на основе алгоритма. Разговорные навыки могут покрывать какую-то определенную тематику диалога, а могут быть способны поддерживать диалог на любые темы. Так, в диалоговой системе открытого домена могут быть скомбинированы навыки открытого домена и навыки закрытого домена, например, для обсуждения фильмов или музыки. Модульная архитектура диалоговых систем в рамках библиотеки DeepPavlov Agent позволяет сочетать навыки открытого и закрытого домена в рамках одной диалоговой системы.

С точки зрения алгоритма, выделяют следующие виды разговорных навыков: шаблонные, ранжирующие, генеративные. Генеративные и ранжирующие навыки обычно используют модели машинного или глубокого обучения, в то время как шаблонные навыки применяют различные эвристики, шаблоны и правила, описанные вручную разработчиками. Каждый вид навыков имеет свои преимущества и недостатки.

Ранжирующие навыки

Одними из самых легко реализуемых, и потому довольно популярных, являются ранжирующие навыки. Для их построения и обучения достаточно иметь набор диалоговых данных соответствующего домена без какой-либо дополнительной разметки. Основная идея ранжирующих моделей в том, чтобы подобрать наиболее подходящую реплику к заданному контексту среди всех или частично отфильтрованных реплик из имеющегося набора диалоговых данных. Алгоритм работы ранжирующих моделей состоит в вычислении меры соответствия контекста и реплик из базы возможных ответов и выбора наиболее подходящего ответа с оптимальным значением меры. Вычисление меры соответствия же может быть сделано двумя способами: (1) получение независимых векторных представлений контекстов и реплик, из которых вычисляется мера

соответствия с помощью специальной функции, (2) «склеивание» контекстов и реплик, которое обрабатывается моделью для получения меры соответствия.

Важными недостатками ранжирующих навыков являются ограниченный контекст диалога, ограниченная вариативность реплик, невозможность автоматической адаптации частей реплики в соответствии с контекстом. Ограниченность контекста возникает в виду того, что популярные на данный момент нейросетевые модели не могут принимать больше определенного количества токенов на вход, а значит длина передаваемого контекста строго ограничена, и при выборе реплики навык может не учитывать информацию, которая была сказана в рамках текущего диалога. Недостаток адаптивности реплик заключается в том, что детали в репликах могут не соответствовать контексту при том, что общая тема и направление дискуссии соответствует контексту, например, если в базе ответов есть реплики с обсуждением различных фильмов, а в контексте упоминается другой фильм. В диалоговой системе DREAM используются несколько ранжирующих навыков: тематические, основанные на TF-IDF векторизации контекстов и реплик, и навык `ConveRT Reddit`. Тематические навыки используют подвыборки диалоговых данных закрытого домена для поддержания диалога на определенные темы, однако зачастую они выдают релевантные, но не соответствующие контексту реплики.

В диалоговой системе DREAM `ConveRT Reddit` использует векторные представления предложений, получаемые с помощью модели ConveRT [48]. Модель выбирает подходящие гипотезы среди заданного набора, ранжируя пары ответ-контекст по косинусному сходству соответствующих векторных представлений. Контекст формируется путем объединения реплик из истории диалога. Модель была предварительно обучена на 654 миллионах пар реплика-ответ. Командой DREAM для участия в конкурсах «Alexa Prize Challenge 3» и «Alexa Prize Challenge 4» модель была также дообучена на 80 тысячах комментариев с сайта Reddit, которые и использовались в качестве базы ответов для выбора реплики-кандидата.

Генеративные навыки

Для обучения генеративных моделей в базовом варианте, когда на вход нейросети подается только векторное представление контекста, а на выходе получается готовая реплика-кандидат, также требуется только наличие набора диалоговых данных. Однако по сравнению с ранжирующими моделями, выход генеративных моделей менее контролируемый – все возможные реплики-кандидаты от ранжирующих моделей представлены в базе диалогов, а генеративные модели могут выдавать абсолютно любые реплики. И если автоматическая фильтрация реплик может быть применена и к базе ранжирующих моделей, и к выходу генеративных моделей, то любая другая, особенно ручная, проверка реплик генеративных моделей не представляется возможной. На данный момент наибольшую популярность среди генеративных моделей имеют языковые модели, предобученные на больших корпусах текстов для решения задачи моделирования естественного языка. В связи с этим, например, генеративные модели на базе языковых моделей даже после дообучения на закрытом домене не могут гарантировать того, что сгенерированная реплика будет принадлежать заданному домену. Поэтому использование генеративных навыков на реальных пользователях является сложной задачей. В конкурсе «Alexa Prize Challenge 3», команда DREAM проводила эксперименты с генеративными моделями TransferTransfo [49], однако в связи с низким качеством они не были включены в итоговую версию системы.

Важным недостатком генеративных навыков является ограниченная вариативность ответов и отсутствие контроля над ходом диалога. Одним из способов повысить контроль над генеративной моделью является использование дополнительной информации, подаваемой на вход модели. Например, это может быть описание персоны бота или другая текстовая информация. Тогда модель учится генерировать реплику на основе контекста и обуславливаться на заданное знание, что позволяет в некотором роде контролировать то, что будет сказано. Такие модели безопаснее применять на реальных пользователях, поэтому в конкурсе «Alexa Prize Challenge 4» в диалоговой системе DREAM использовался навык `Knowledge Grounding Skill`, основанный на нейросетевой генеративной модели ParlAI Blender 90M [50], дообученной на наборе диалогов Topical Chat Enriched [51] и принимающей на вход текущую реплику пользова-

теля, историю диалога и несколько предложений дополнительной информации. Именно за счет подаваемой дополнительной информации и становится возможным ограниченный контроль за тем, о чем пойдет речь в реплике. Тем не менее, навык `Knowledge Grounding Skill` зачастую генерировал реплики-кандидаты, которые не учитывали переданное знание (подробнее про эксперименты с генеративными моделями в техническом отчете DREAM [24]).

Шаблонные навыки

Во многих случаях требуется предсказуемость поведения диалоговой системы, например, при использовании ее непосредственно на реальных пользователях. Поведение системы должно быть не просто предсказуемым, а в некоторых случаях строго определенным. Например, если пользователь спрашивает бота, участвующего в конкурсе «Alexa Prize Challenge», о его имени или команде-создателе, диалоговая система должна не просто не называть свои данные, но должна отказаться раскрывать свои персональные данные по правилам конкурса. Или например, в случае, когда пользователь спрашивает медицинского или финансового совета, бот также должен отказаться от помощи пользователю в связи с недостаточной компетентностью в этом вопросе. В таких случаях самыми подходящими для использования являются навыки шаблонные навыки (так как реплики-ответы обычно строятся на основе шаблонов), которые используют различные правила, эвристики и шаблоны для понимания контекста и генерации ответа.

Одним из самых популярных примеров шаблонных навыков является ELIZA – диалоговая система, проверяющая реплики пользователя на соответствие шаблонам и выдающая соответствующий шаблонный ответ. Само устройство ELIZA положило начало инструменту создания шаблонных навыков – языку разметки искусственного интеллекта (англ.: Artificial Intelligence Markup Language, AIML). Этот инструмент будет рассмотрен подробнее в Разделе 1.4.2. Другим примером являются задаче-ориентированные разговорные навыки, например, для совершения покупок, заказа услуг или ответа на часто задаваемые вопросы. Это связано со структурой задаче-ориентированных диалоговых систем, в которых модуль понимания реплики пользователя заклю-

чается в распознавании намерений и извлечении слотов, а ответ выбирается в соответствии с заданными в алгоритме правилами среди шаблонных вариантов, в котором также заполняются значения слотов. Причем заполнение слотов может происходить не только за счет информации из реплики пользователя, но и с помощью графов знаний. Соответственно, любой навык, выбор реплики в котором происходит за счет интерпретируемого алгоритма, основанного на шаблонах, извлеченной из контекста или полученной из графов знаний информации, представляет из себя шаблонный навык.

Сценарные навыки

Самым важным недостатком ранжирующих и генеративных моделей, который фактически не может быть гарантированно решен, является недостаток контроля развития диалога. Данная проблема может быть решена в рамках шаблонных навыков за счет изначальной проработки сценария диалога на несколько ходов. Что приводит к введению понятия *сценарных навыков* – это навыки, в которых заложена структура хода диалога на несколько шагов – *сценарий*. Каждый шаг в сценарии обычно включает проверку реплики пользователя на соответствие контексту, обработку извлеченной информации, переход к следующей реплике сценария за счет набора условий и пост-обработка (например, заполнение слотов). Основное достоинство сценарных навыков состоит в том, что они позволяют создать впечатление связного диалога «в глубину» в течение сразу нескольких шагов диалога. Сценарии также позволяют в некотором роде интегрировать персону бота в диалог: не только предпочтения, но и личные истории, события. Большинство задаче-ориентированных навыков также являются сценарными (в случаях, когда задача выполняется больше, чем за один шаг диалога), так как им требуется несколько шагов для сбора необходимой информации для выполнения задачи.

При этом не каждый шаблонный навык обязательно следует сценарию – шаблонные навыки вполне могут иметь одношаговые алгоритмы, а значит не все шаблонные навыки являются сценарными. С другой стороны, в рамках сценарного навыка вполне можно использовать ранжирующие или генеративные модели для построения ответа на некоторых шагах сценария. Так, например,

`Movie Skill` в рамках сценария обсуждения конкретного фильма при наличии в базе краткого описания сюжета, может сгенерировать с помощью `Knowledge Grounding Service` реплику, которая обусловлена на краткое описание сюжета. Смысл генерируемой реплики сложно контролировать, поэтому чтобы решить эту проблему, был использован следующий подход: в качестве контекста в генеративную модель подается специальный вопрос и в качестве дополнительной информации необходимое для ответа знание. Например, использование специального вопроса «What is your favourite moment?» («Какой у тебя любимый момент?») и краткого описания сюжета в качестве дополнительной информации позволяет с большой уверенностью получить в качестве ответа от системы реплику, описывающую некоторый момент из сюжета, что позволяет встроить эту реплику в сценарий в рамках обмена мнениями о любимом моменте фильма.

Тем не менее, сценарные навыки также обладают некоторыми упомянутыми выше недостатками. Во-первых, сценарные навыки обладают ограниченным лексическим разнообразием, так как большинство реплик в сценариях являются шаблонными. Во-вторых, ограниченность покрываемых ситуаций – в большинстве случаев сценарии не обрабатывают проявление инициативы пользователем, что может создать негативное впечатление того, что бот не слышит пользователя. Дополнительную сложность создает разработка сценарных навыков, которая подразумевает создание и обработку большого количества правил и условий для прохода по сценарию. Эта проблема по большей части решается за счет грамотно подобранных инструментов для построения разговорных навыков, которые будут рассмотрены в Разделе 1.4.2.

1.4.2 Инструменты для построения разговорных навыков

Благодаря использованию контейнерной парадигмы в DeepPavlov Agent есть возможность использовать навыки, построенные с помощью любых инструментов, в том числе встраивать в качестве навыков отдельные диалоговые системы. Это значительно облегчает возможность интеграции готовых решений в текущую диалоговую систему.

Язык разметки искусственного интеллекта (англ.: Artificial Intelligence Markup Language, AIML) – это диалект XML (англ.: eXtensible Markup

Language) для создания диалоговых агентов. AIML – это хорошо документированный, широко распространенный и простой в использовании язык для реализации диалоговых систем, которые основаны на шаблонной проверке реплик пользователя. Поэтому навыки на основе AIML могут быть использованы в качестве разговорных навыков.

В связи с недостаточно устоявшимися стандартами инструментов построения диалоговых систем, многие компании разрабатывают и используют собственные инструменты построения диалоговых систем, так называемые доменно-специфичные языки (англ.: Domain Specific Languages, DSL). Например, DSL на основе библиотеки DeepPavlov[17].

По результатам конкурса «Alexa Prize Challenge 3» в 2020 году команда Emora опубликовала в открытый доступ[18] фреймворк STDM (англ.: State Transition Dialogue Manager) [52] для создания сценарных диалоговых систем. Команда Emora использовала данный фреймворк для покрытия большого числа популярных тем сценариями, и заняла первое место в конкурсе «Alexa Prize Challenge 3». Фреймворк позволяет строить диалоговый граф при помощи обработки реплики пользователя шаблонами, регулярными выражениями, условиями на намерения и сущности.

Основную сложность в применении STDM в качестве навыка в рамках библиотеки DeepPavlov Agent представляет хранение состояния диалога внутри STDM модели, что противоречит концепции DeepPavlov Agent, согласно которой вся информация о диалоге находится внутри состояния диалога, которое хранится в отдельной базе данных и передается между компонентами, не хранящими внутри себя информацию о конкретных диалогах и пользователях. Поэтому во время конкурса «Alexa Prize Challenge 4» команда DREAM работала над адаптацией фреймворка STDM к применению в качестве навыка в рамках DeepPavlov Agent. В результате был опубликован в открытый доступ[19] фреймворк `Dialog Flow Framework` (`DFF`). Фреймворк `DFF` был активно использован командой DREAM для создания сценарных навыков, покрывающих различные популярные темы, а также расширен до возможности автоматической интеграции знаний с Wikipedia.

[17] http://docs.deeppavlov.ai/en/master/features/skills/dsl_skill.html

[18] https://github.com/emora-chat/emora_stdm

[19] https://github.com/deepmipt/dialog_flow_framework

1.5 Проблемы диалоговых систем

Классификация является одной из самых базовых задач обработки естественного языка. Большой прорыв в решении задачи был сделан после появления векторных представлений word2vec [53; 54], GloVe [55] и позднее fastText [40]. Качество решения задач обработки естественного языка значительно выросло при использовании предобученных векторных представлений, что привело к дальнейшим исследованиям области, направленным на улучшение качества векторных представлений. Одними из основных параметров моделей векторных представлений являются их язык и домен – область происхождения исходных текстов, на которых была обучена модель. Это привело к появлению двух основных направлений исследований векторных представлений: обучение многоязычных моделей и улучшение качества одноязычных моделей путем спецификации обучающей выборки. В 2018 году начала активно развиваться область решения задачи языкового моделирования, в которой глубокие нейронные сети большого размера обучаются предсказывать следующий элемент текста (символ, токен или целое предложение). Появление методологии получения контекстно-зависимых векторных представлений из языковых моделей [3] (англ.: Embeddings from Language Models, ELMo) в 2018 году положило начало новым исследованиям векторных представлений. Появление архитектуры Трансформер [56] (англ.: Transformer) позволило сделать большой прорыв в языковом моделировании. Сейчас лучшие результаты на многих задачах демонстрируют модели [5; 57—63] на базе архитектуры Трансформер. В Главе 2 представлено исследование влияния языкового стиля векторных представлений на качество решения задачи классификации текстов.

В связи с серьезным прорывом в области обработки естественного языка, вызванным достижениями предобученных языковых моделей, и нарастающей цифровизацией человеческой жизни, популярность диалоговых систем начала стремительно возрастать. Диалоговые системы для ответов на часто-задаваемые вопросы на различных веб-сайтах и персональные голосовые помощники прочно вошли в жизни обычных людей. На текущий момент в индустрии в основном используются задаче-ориентированные диалоговые системы, однако популярностью пользуются и чат-боты для обычного общения – «Replika», «Xiaoice» [11] и менее известные системы завоевывают рынок. Однако полно-

ценный тест Тьюринга на текущий момент еще не пройден ни одной системой в мире, а пользователи при общении с диалоговыми системами зачастую испытывают чат-бота на знание и понимание окружающего мира. Тем не менее, большинство диалоговых систем не обладают достаточной эмпатичностью и осознанностью, чтобы их реплики имели человеческий вид. Пользователи колонок «Amazon Alexa» знают, что разговаривают со встроенной диалоговой системой, поэтому недостаточно «человечно» звучащие фразы бота воспринимаются лояльно. Более того, по результатам анализа [64], внедрение ботом в разговор интересных фактов помогает улучшить пользовательский опыт и не создает впечатление излишне нагруженного информацией диалога.

Интеграция фактологической информации не ограничивается использованием только интересных фактов, так как формулируемые в виде полноценных утверждений факты не дают достаточной гибкости использования баз знаний. Структурированная информация может быть использована для анализа сущностей, извлекаемых из реплики пользователя, поиска их связей с другими сущностями, заполнения слотов в шаблонных репликах. Интеграция баз знаний в таком виде значительно автоматизирует сценарные навыки, которые могут автоматически проверять поступающую информацию, не только классифицируя темы и намерения пользователей, но и анализируя получаемые сущности, их происхождение и связи. Также сценарные навыки могут формировать ответные реплики, заполняя шаблоны упоминаемыми или связанными сущностями и отношениями. В Главе 3 описано общее устройство диалоговой системы DREAM и, в частности, интеграция баз знаний в нее.

Другим важным источником знаний является здравый смысл, подразумевающий знания, которыми люди обладают изначально и не ищут в базах знаний. Несмотря на значительный прогресс языковых моделей во многих задачах обработки естественного языка, некоторые системы по-прежнему плохо справляются [14; 15] с ответами, основанными на здравом смысле. Аналогично, даже при использовании специальных вопросно-ответных систем для ответов на фактоидные вопросы, некоторые чат-боты зачастую не могут отвечать на простые вопросы об окружающем мире. Поэтому в Главе 4 представлено исследование интеграции моделей предсказания здравого смысла в диалог.

Представленные в Главе 4 навыки, интегрирующие здравый смысл, являются сценарными. Ранжирующие и генеративные навыки по-прежнему не могут гарантировать соответствие контексту и непротиворечие здравому

смыслу возвращаемых реплик. При этом, ранжирующие навыки являются достаточно ограниченными в языковом разнообразии, а также не позволяют универсальным образом использовать реплики, которые можно было бы обобщить простой заменой. Более универсальные генеративные навыки меньше поддаются контролю, в связи с полной непредсказуемостью языковой модели. Тем не менее, предлагаются различные решения для контроля генерации нейросетевых моделей для применения в диалоговых системах, например, идея условной генерации [65] реплик, заключающаяся в том, что нейросеть продолжает реплику после заданного шаблона с заполненными слотами. Другой вариант контроля состоит в подаче дополнительной информации в нейросеть для обусловленной генерации.

Однако даже такие подходы не позволяют в достаточной степени контролировать выдачу генеративных моделей, в связи с чем возникают сложности первой части диалогового менеджера в рамках фреймворка DeepPavlov Agent. Так, например, в некоторых ситуациях от диалоговой системы требуется определенное поведение, поэтому выбор навыков для генерации реплик-кандидатов в диалоговой системе DREAM является алгоритмом, основанным на различных условиях на аннотации. Выбор финального ответа среди реплик-кандидатов представляет из себя более сложную задачу: показатели уверенности навыков имеют разное происхождение, необходимо учитывать пассивных и проактивных пользователей, нужна приоритизация сценариев для связности диалога «в глубину». Для более широкого покрытия тем и ситуаций нельзя использовать только алгоритмы выбора реплики, основанные на различных условиях на аннотации, но и нет возможности полагаться исключительно на ранжирующие нейросетевые модели. Подробное описание алгоритмов выбора навыков и выбора финальной реплики в диалоговой системе DREAM, а также эксперименты с алгоритмом выбора финального ответа представлены в Главе 5.

Нельзя не отметить саму сложность оркестрирования большой диалоговой системой, имеющей десятки компонент разного происхождения и разного уровня потребления ресурсов. Для устойчивости системы к постоянном потоку реальных пользователей также требуется дублирование некоторых компонент и грамотное распределение нагрузки на отдельные контейнеры. Библиотека с открытым исходным кодом DeepPavlov Agent[20] для построения диалоговых

[20] https://github.com/deepmipt/dp-agent

систем существенно помогла упростить процесс разработки и поддержания диалоговой системы DREAM.

Глава 2. Классификация текстов разговорного домена

Классификация последовательностей является одной из важнейших задач при построении модульных диалоговых систем, так как фактически покрывает целый список задач: классификация тем, распознавание намерений, анализ тональности, теггирование (классификация элементов) последовательности, которое в свою очередь покрывает извлечение сущностей и определение их типов, в некоторых случаях даже рекомендательные системы. Автор участвовала в разработке моделей классификации текстов для библиотеки DeepPavlov [19], а также обучала и интегрировала модели классификации в диалоговую систему DREAM [21; 24], участвующую в конкурсах «Alexa Prize Challenge 3» и «Alexa Prize Challenge 4».

Однако классификация текстов, как и любые другие задачи понимания естественного языка, в контексте диалоговых систем имеет особенности, связанные с доменной специфичностью, а именно со стилистикой разговорной речи [66]. В данной главе автор описывает базовые нейросетевые методы классификации текстов в Разделе 2.1, стилистическую специфичность векторных представлений моделей `fastText` и языковых моделей ELMo в Разделе 2.2, а также самых популярных на текущий момент векторных представлений языковых моделей BERT в Разделе 2.3.

Автором были разработаны и обучены векторные представления `fastText` и модели классификации, предложенные в данной главе. Векторные представления языковых моделей ELMo и BERT были обучены соавторами работ.

2.1 Базовые нейросетевые методы

В данном разделе 2.1 представлено решение задачи классификации текстов на основе нейросетевых методов, предложены способы повышения качества с помощью изменения архитектуры сети, проведено сравнение результатов с другими готовыми системами распознавания намерения пользователей. Исследование проведено на наборе данных SNIPS[1], содержащем 2400 примеров для

[1] https://github.com/snipsco/nlu-benchmark/tree/master/2017-06-custom-intent-engines

каждого из 7 заданных намерений. На наборе комментариев с сайта Reddit[2] с использованием библиотеки `fastText` [40] обучена модель получения векторных представлений слов.

Исследование было представлено в докладе [18], архитектуры и векторные представления использованы в статье [19].

В качестве базовой модели используется неглубокая широкая (англ.: shallow-and-wide, SWCNN) свёрточная нейронная сеть, оригинально предложенная в [67]. Векторизованные данные поступают на вход нейронной сети, состоящей из трех сверточных слоев с разными размерами ядра и слоев глобальной субдискретизации (англ.: global max pooling). Полученные вектора далее конкатенируются и передаются на вход полносвязным слоям. Архитектура сети представлена на рисунке 2.1.

Другая архитектура получается добавлением в базовую модель дополнительных признаков, таких как векторные представления запроса целиком (англ.: *sentence embeddings*), получаемых с помощью предобученной модели InferSent [68]. Отличие данной нейросетевой модели в том, что полученное векторное представление запросов целиком конкатенируется с выходом слоев глобальной субдискретизации.

Еще одна модель добавляет возможность комбинации задач классификации текстов и распознавания именованных сущностей. На вход нейросети также подаются векторные представления именованных сущностей, упомянутых в тексте, полученные с помощью BiLSTM-CRF (англ.: Bidirectional Long-Short Term Memory with a Conditional Random Fields) архитектуры модели распознавания именованных сущностей [69], которые также конкатенируются с выходом слоев глобальной субдискретизации.

Рекуррентные нейронные сети зачастую показывают результаты лучше, чем свёрточные сети в задачах обработки текста, в связи с чем создана модификация модели, в которой свёрточные слои были заменены двунаправленной моделью с долгой краткосрочной памятью BiLSTM (англ.: Bidirectional Long-Short Term Memory) [70]. Архитектура модели представлена на рисунке 2.2.

Результаты представлены в Таблице 1 и демонстрируют преимущества неглубокой широкой свёрточной сети (в таблице обозначена как CNN) над двунаправленной моделью с долгой краткосрочной памятью (в таблице обозначена как BiLSTM) для рассматриваемой задачи классификации текстов. Базовая

[2]Набор данных с сайта Reddit «RC_2011-01»: http://files.pushshift.io/reddit/comments/

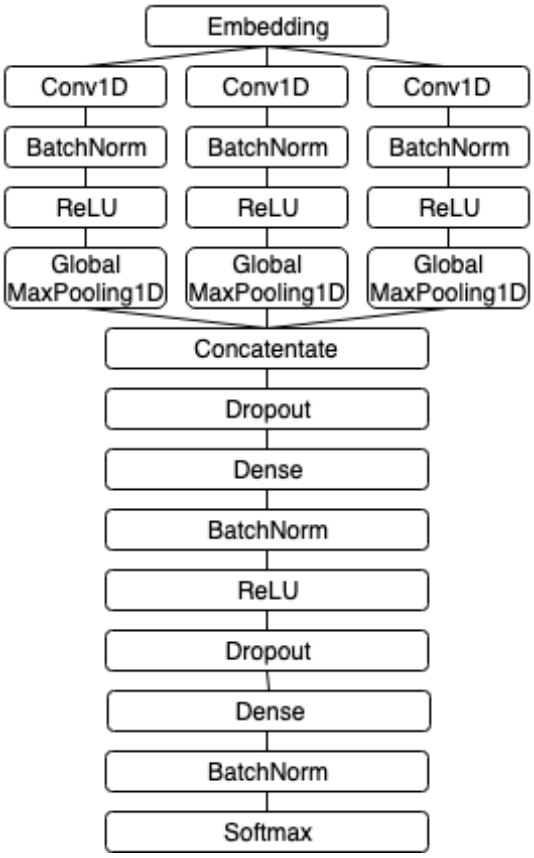

Рисунок 2.1 — Неглубокая широкая свёрточная нейронная сеть (shallow-and-wide, SWCNN).

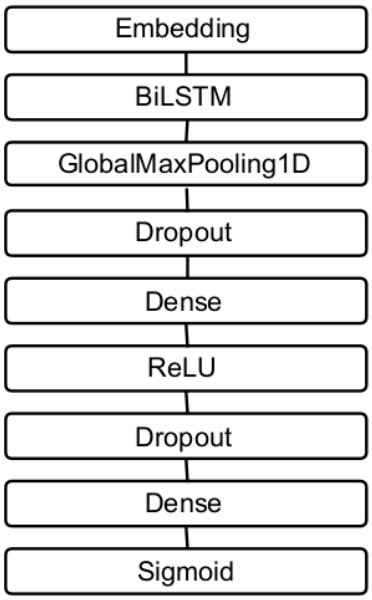

Рисунок 2.2 — Рекуррентная нейронная сеть с двунаправленной долгой краткосрочной памятью (Bidirectional Long-Short Term Memory, BiLSTM).

неглубокая широкая свёрточная сеть с подобранными параметрами показывает результаты лучше, чем готовые сервисы-модели[3], включая *api.ai* и *ibm.watson*. Однако стоит учитывать, что готовые сервисы-модели обучались в формате черного ящика (англ.: *black box*), а значит, подбор параметров мог быть проведен только внутри самой системы автоматически, в то время как в текущей работе подбор параметров производился с помощью оценки результатов на валидационной выборке.

[3] `https://www.slideshare.net/KonstantinSavenkov/nlu-intent-detection-benchmark-by-intento-august-2017`

Модель	Среднее значение F-меры для каждого намерения						
	Add To Playlist	Book Restaurant	Get Weather	Play Music	Rate Book	Search Creative Work	Search Screening Event
AA[3]	0.9931	0.9949	0.9935	0.9811	0.9992	**0.9659**	**0.9801**
IW[3]	0.9931	**0.9950**	**0.9950**	**0.9822**	**0.9996**	0.9643	0.9750
ML[3]	**0.9943**	0.9935	0.9925	0.9815	0.9988	0.9620	0.9749
WA[3]	0.9877	0.9913	0.9921	0.9766	0.9977	0.9458	0.9673
SA[3]	0.9873	0.9921	0.9939	0.9729	0.9985	0.9455	0.9613
RA[3]	0.9894	0.9943	0.9910	0.9660	0.9981	0.9424	0.9539
AL[3]	0.9930	0.9862	0.9825	0.9709	0.9981	0.9427	0.9581
CNN	0.9956	**0.9973**	0.9968	0.9871	**0.9998**	0.9752	**0.9854**
CNN & NER	**0.9964**	0.9958	0.9920	0.9865	0.9970	0.9652	0.9768
CNN & InferSent	0.9956	0.9971	**0.9969**	**0.9879**	0.9994	**0.9753**	0.9845
BiLSTM	0.9612	0.9488	0.9514	0.9351	0.9688	0.8769	0.8979
CNN & NER-truth	0.9996	0.9987	0.9986	0.9997	1.0000	1.0000	1.0000

Таблица 1 — Результаты экспериментов. Названия моделей сокращены: AA - api.ai, IW - ibm.watson, ML - microsoft.luis, WA - wit.ai, SA - snips.ai, RA - recast.ai, AL - amazon.lex. Результаты в верхней части таблицы (метрики для сторонних моделей) получены не автором.

2.2 Векторные представления ELMo для классификации текстов

Векторные представления языковых моделей (англ.: Embeddings from Language Models, ELMo) [71] – это векторные представления, получаемые из двунаправленной рекуррентной модели с долгой краткосрочной памятью BiLSTM, обученной для решения задачи языкового моделирования на большом текстовом корпусе. Векторные представления ELMo глубокие и контекстно-зависимые. Вектора-выходы с внутренних слоев сети можно комбинировать и

использовать аналогично другим векторным представлениям токенов, таким как `fastText`[40], однако в данном случае вектора обладают важным свойством – представление каждого слова формируется и левым, и правым контекстами этого слова. Для обучения языковых моделей требуются большие текстовые корпуса и значительные вычислительные ресурсы.

В русскоязычном сообществе исследователей обработки естественного языка на момент публикации оригинальной статьи [20] велись активные дискуссии о реальной производительности ELMo, было оставлено множество отрицательных отзывов о качестве нейросетевых моделей, основанных на языковых моделях или использующих векторные представления ELMo. Таким образом, основная цель данного раздела 2.2 – сравнить качество нейросетевых моделей, использующих векторные представления `fastText` и ELMo моделей, обученных на корпусах с разными языковыми стилями.

Исследование, представленное в данном разделе, было опубликовано в статье [20].

2.2.1 Данные для задачи языкового моделирования

Русскоязычные языковые модели для научного стиля обучены на наборах текстов с веб-сайта Wikipedia[4], публицистического – Russian WMT News[5], а русскоязычные языковые модели для разговорного стиля обучены на наборе постов с сайта Twitter[6]. Ключевые характеристики наборов данных представлены в Таблице 2.

Наборы данных News WMT доступны для скачивания уже в предобработанном и очищенном виде, данные Wikipedia были почищены от html-разметки, а в наборе данных Twitter все хэштеги и логины пользователей заменены на специальные токены. Размер словаря для каждого набора данных составляет 1 миллион самых частотных токенов. Каждый набор данных разделен на обучающую (98%) и валидационную (2%) выборки.

[4] https://ru.wikipedia.org/
[5] http://www.statmt.org/
[6] https://twitter.com/

Данные	Количество слов	Размер словаря	Среднее число слов в предложении	Размер файла
Wikipedia	472 M	5.6 M	19.4	4.8 Gb
WMT News	1133 M	4.1 M	19.6	12.0 Gb
Twitter	887 M	11.3 M	8.7	7.9 Gb

Таблица 2 — Ключевые характеристики наборов данных, на которых обучались языковые модели.

2.2.2 Данные для задачи классификации

Датасет RuSentiment был опубликован в 2018 году [72] вместе с базовыми результатами. Полный набор данных содержит более 30 тысяч сообщений из социальной сети Вконтакте, средняя длина сообщения 17 токенов, каждый пост соотнесен с одним из классов: позитивная, негативная и нейтральная тональности, речевые выражения и пропуск. В данной работе использовалась подвыборка «случайных сообщений» (англ.: «random posts»), которая также была поделена на обучающую и валидационную выборки в соотношении 9 к 1. Подвыборка «предварительно выбранных сообщений» (англ.: «pre-selected posts») в данной работе не используется. Тестовая выборка такая же, как и в исходной статье.

Лингвисты выделяют пять стилей русского языка: научный, официальный, публицистический, художественный и разговорный. Первые четыре стиля и последний сильно различаются с точки зрения лексики и морфологии. Поэтому датасет RuSentiment и был выбран в качестве целевого набора данных, так как его содержание соответствует разговорному стилю, который ранее зачастую не включался в данные для языкового моделирования, в то время как использование разговорных данных в 2019 году как раз начало набирать популярность в связи с ростом популярности диалоговых систем.

2.2.3 Предобучение языковых моделей и векторные представления

В данном разделе исследуются следующие векторные представления, покрывающие различные языковые стили:

- векторные представления `fastText`, обученные на Russian Wikipedia и WMT News корпусах,
- векторные представления `fastText`, обученные на Russian Twitter corpus,
- векторные представления ELMo, предобученные на датасете Russian WMT News,
- векторные представления ELMo, предобученные на датасете Russian Wikipedia,
- векторные представления ELMo, предобученные на датасете Russian Twitter,
- векторные представления ELMo, предобученные на датасете Russian Twitter и дообученные на датасете RuSentiment.

Векторные представления `fastText` размерностью 300 были обучены с параметрами по умолчанию обучения skipgram модели [40], принимающей на вход n-граммы длинной от 3 до 6 символов. Русскоязычные векторные представления `fastText` размерности 300 не разговорного языкового стиля обучены на датасетах Russian Wikipedia и Russian WMT News, а русскоязычные векторные представления `fastText` разговорного стиля размерности 300 обучены на датасете Twitter. Обе модели `fastText` выложены в открытый доступ[7].

Языковая модель ELMo состоит из двух основных частей: свёрточных слоев и 2 блоков из 2 рекуррентных слоев. В исходной реализации модель получает на вход индексы символов в кодировке utf-8 (от 0 до 255 и три специальных символа для дополнения до нужной длины, а также обозначающие начало и конец слова). В каждый из рекуррентных блоков передаются представления из свёрточных слоев, при чем каждый блок обрабатывает представления в своем направлении аналогично двунаправленной рекуррентной модели.

Обучение проводилось в аналогичной [73] и [74] манере. Дополнительный полносвязный слой, за которым следует слой softmax, используется для

[7] http://docs.deeppavlov.ai/en/master/features/pretrained_vectors.html#fastText

обучения языковой модели. Этот слой не используется в дальнейшем при получении векторных представлений из предобученной языковой модели. Для получения самих контекстно-зависимых векторных представлений используется взвешенная сумма векторных представлений со всех промежуточных слоев нейросети. Коэффициенты для этой суммы могут быть обучены, а значит, могут быть разными при решении различных задач. Последний слой также можно использовать аналогично TagLM [75] и CoVe [76]. Векторные представления предложений целиком обычно формируются как среднее или взвешенная с коэффициентами из TF-IDF сумма [77] векторов слов.

В данной работе используется модель 4096/512 с 93.6 миллионами параметров[8]. Результаты обучения языковых моделей на предложенных датасетах Wikipedia, WMT News, Twitter и дообучения языковой модели Twitter на RuSentiment представлены в Таблице 3. Все языковые модели обучались 10 эпох, каждая на 3 видео-картах 1080ti. Дообучение производилось до тех пор, пока перплексия на валидационной выборке возрастала. Итоговая перплексия языковой модели на выборке «random posts» датасета RuSentiment составляет 159.2 и была достигнута после 4 эпох обучения. Предобученные языковые модели были также протестированы на всей выборке «random posts» датасета RuSentiment, результаты чего представлены в последней колонке Таблицы 3. Языковая модель, обученная на датасете Twitter показывает лучшие результаты, как и ожидалось, в связи с совпадением стилистических доменов. Полученные языковые модели ELMo выложены в открытый доступ[9].

Данные	Время обучения	Эпохи	Перплексия на валид.	Перплексия на RuSentiment
Wiki	6 дней	10	43.692	17364.89
WMT News	14 дней	10	49.876	360.97
Twitter	10 дней	10	94.145	172.25
Дообученная Twitter на RuSentiment	15 минут	4	159.2	–

Таблица 3 — Результаты обучения и дообучения языковых моделей ELMo.

[8] https://allennlp.org/elmo

[9] http://docs.deeppavlov.ai/en/master/features/pretrained_vectors.html#elmo

2.2.4 Обучение моделей классификации

На момент публикации оригинальной статьи [20] наиболее распространенными подходами к нейросетевой классификации текстов были свёрточные и рекуррентные сети. Поэтому в данной работе в качестве базовых архитектур используются неглубокая широкая сверточная сеть (shallow-and-wide, SWCNN) [78] и двунаправленная GRU (Bidirectional Gated Recurrent Unit, BiGRU) [79; 80].

Модель SWCNN подробно описана в разделе 2.1 и изображена на рисунке 2.1, а архитектура модели BiGRU представлена на рисунке 2.3. Необучаемые векторные представления попадают на вход двунаправленному GRU слою, за которым следуют слои глобальной субдискретизации (англ.: global max and average pooling). Полученные вектора конкатенируются с двумя последними состояниями из BiGRU и передаются в полносвязные слои.

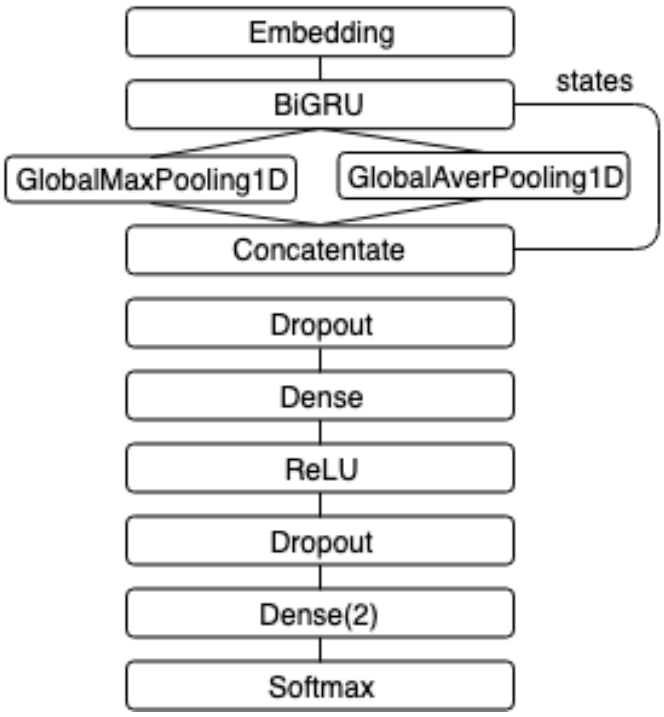

Рисунок 2.3 — Используемая BiGRU архитектура.

Представленные базовые нейросетевые модели обучены на векторных представлениях модели `fastText` размерности 300 и языковых моделей ELMo размерности 1024 различного языкового стиля. Целевая метрика – взвешенная F1, обучение проводилось до начала ухудшения значений целевой метрики на

валидационной выборке. Все эксперименты имеют одинаковые параметры, свёрточные слои содержат 256 фильтров и ядра размеров 3, 5, 7, а размер BiGRU слоя равен 256. Первый полносвязный слой имеет размер 100 для обеих сетей. Лимит ухудшения значения целевой метрики на валидационной выборке по количеству эпох равен 2, а максимальное число эпох равно 10. Для модели SWCNN используется регуляризация в виде dropout rate равного 0.5, L2-коэффициенты 10^{-3} и 10^{-2} для сверточных и полносвязных слоев соответственно. Для модели BiGRU dropout rate равен 0.2, L2-коэффициент равен 10^{-6} для рекуррентных и полносвязных слоев.

Каждый эксперимент проводился 4 раза, в результате средние взвешенные F1-метрики представлены в Таблице 4. Для векторных представлений `fastText` модель BiGRU показывает результаты лучше, чем SWCNN, в то время как свёрточные модели с использованием векторных представлений ELMo превосходят рекуррентные. Модели, использующие векторные представления научного и публицистического стилей достигают результатов, сравнимых с исходной статьей [72] (71,7 взвешенная F1-метрика без использования «pre-selected posts»). Обе архитектуры модели, использующие векторные представления `fastText` разговорного стиля (обученные на наборе данных Twitter) превосходят не только базовые результаты из исходной статьи [72], но и все модели, обученные на векторных представлениях научного (Wikipedia) и публицистического (WMT News) стилей. Классификаторы, использующие векторные представления языковой модели разговорного стиля, превосходят по качеству все рассматриваемые модели. Лучшие результаты (почти на 6 пунктов выше, чем у предыдущих опубликованных результатов) достигнуты неглубокой широкой свёрточной нейросетью, использующей векторные представления языковой модели ELMo, обученной на датасете Twitter.

2.3 Векторные представления BERT для классификации текстов

Появление первых работ по применению языковых моделей [3] можно назвать прорывом в области задач обработки естественного языка. Было положено начало целому направлению исследований, благодаря которому в дальнейшем появилась знаменитая архитектура Трансформер [56]. Сейчас язы-

Модель	Вектора	F1-weighted на валид.	F1-weighted на тест.
Rogers et al. [72]	`fastText` VK	–	72.8
SWCNN	`fastText` Wiki+News	67.84	70.27
BiGRU	`fastText` Wiki+News	69.54	71.74
SWCNN	`fastText` Twitter	70.91	73.03
BiGRU	`fastText` Twitter	72.62	74.45
SWCNN	ELMo WMT News	70.27	72.42
BiGRU	ELMo WMT News	70.15	71.37
SWCNN	ELMo Wiki	68.11	71.28
BiGRU	ELMo Wiki	66.55	69.47
SWCNN	ELMo Twitter	75.40	**78.50**
BiGRU	ELMo Twitter	**75.89**	77.62
SWCNN	ELMo Fine-tuned	74.74	77.98
BiGRU	ELMo Fine-tuned	75.75	77.19

Таблица 4 — Итоговые значения метрик классификации на датасете RuSentiment для различных векторных представлений.

ковые модели на базе Трансформера по-прежнему показывают одни из лучших результатов в решении многих задач обработки естественного языка. Модель BERT [5] является одной из наиболее известных моделей на базе Трансформеров, поэтому в данном Разделе 2.3 мы рассмотрим влияние языкового стиля на примере решения задач классификации текстов на базе предобученных языковых моделей BERT.

Сотрудниками лаборатории нейронных систем и глубокого обучения МФТИ в рамках работ над библиотекой для создания диалоговых систем DeepPavlov в 2019 году были обучены и выложены в открытый доступ предобученные языковые модели[10] на базе архитектуры BERT, адаптированные под различные домены, такие как *разговорный английский язык, русский язык и разговорный русский язык*. Подробнее о методологии дообучения языковых моделей для других языков и специфичных доменов изложено в диссертации Куратова Юрия[11]. В данной работе представлены результаты исследования вли-

[10] http://docs.deeppavlov.ai/en/master/features/models/bert.html

[11] https://mipt.ru/upload/medialibrary/099/dissertatsiya-kuratov.pdf

яния доменной специфичности языковой модели BERT на качество решения задачи классификации текстов, которое было проведено автором.

Архитектура для решения задачи классификации текстов на базе языковых моделей BERT получается добавлением одного полносвязного слоя, принимающего на вход векторное представление всего текста целиком из модели BERT (в данной работе рассматривается архитектура BERT-base с размерностью скрытого слоя 768), и выдающего вектор, содержащий распределение принадлежности к заданным классам. Векторным представлением текста целиком из модели BERT является выход с последнего слоя нейросети BERT для специального токена *[CLS]*, который был пропущен через еще один полносвязный слой с гиперболическим тангенсом в качестве функции активации. В данной работе во всех моделях дообучались все слои нейросети. На вход полученной нейросетевой модели подаются тексты, разделенные на токены, которые слева дополняются специальным токеном начала строки *[CLS]*.

2.3.1 Данные для задачи классификации текстов

Для оценки влияния языкового стиля языковой модели в данной работе рассмотрим задачи классификации для следующих наборов данных:

- RuSentiment [72] – русскоязычный датасет постов из социальной сети ВКонтакте. Подробнее описан в Разделе 2.2.2.
- Twitter Mokoron [81] – русскоязычный датасет для определения тональности постов из социальной сети Twitter[12]. Содержит разметку на два класса, позитивные и негативные сообщения, которая была получена автоматически с использованием правил.
- Yelp Reviews[13] – англоязычный датасет для определения тональности отзывов с сайта Yelp, содержащий 5 классов, соответствующих целочисленному рейтингу, оставленному пользователем при написании отзыва.
- Kaggle Insults[14] – англоязычный датасет для определения оскорбительный текстов, представленный в рамках соревнования «Detecting Insults

[12] https://www.twitter.com

[13] https://www.yelp.com/dataset/challenge

[14] https://www.kaggle.com/c/detecting-insults-in-social-commentary

in Social Commentary» на Kaggle[15]. Содержит бинарную разметку, является ли комментарий оскорбительным.

– Stanford Sentiment Treebank (SST) [82] – англоязычный датасет для определения тональности отзывов к фильмам. Содержит разбиение на 5 классов: очень позитивные, позитивные, нейтральные, негативные, очень негативные.

2.3.2 Результаты для задачи классификации текстов

В Таблице 5 представлены результаты обучения моделей классификации на основе моделей BERT разных языковых стилей для русского и английского языков. Результирующие значения метрик усреднены по 5 запускам обучения. Модели, обозначенные «BERT», в качестве базовой языковой модели использовали BERT-base English[16] и RuBERT-base[17] для английских и русских датасетов соответственно. Для моделей, обозначенных «Разговорный BERT», в качестве предобученной языковой модели использовались Conversational BERT English[18] и Conversational RuBERT[19] для английского и русского языков соответственно. Использование предобученной модели разговорного языкового стиля для всех рассмотренных наборов данных, содержащих посты, комментарии и отзывы с различных социальных сетей, для английского и русского языков улучшает значение целевой метрики для всех рассматриваемых задач классификации текстов.

По результатам, полученным в этой главе, можно сделать следующие **выводы**:

– Обученные в рамках работы векторные представления `fastText` позволяют улучшить качество классификации текстов по сравнению с базовыми моделями для датасета RuSentiment.

[15] https://www.kaggle.com

[16] http://files.deeppavlov.ai/deeppavlov_data/bert/cased_L-12_H-768_A-12.zip

[17] http://files.deeppavlov.ai/deeppavlov_data/bert/rubert_cased_L-12_H-768_A-12_v2.tar.gz

[18] http://files.deeppavlov.ai/deeppavlov_data/bert/conversational_cased_L-12_H-768_A-12_v1.tar.gz

[19] http://files.deeppavlov.ai/deeppavlov_data/bert/ru_conversational_cased_L-12_H-768_A-12.tar.gz

Данные	Язык	Метрика	BERT	Разговорный BERT
RuSentiment	рус.	F1	72.6	**76.2**
Twitter Mokoron	рус.	Accuracy	99.90	**99.95**
Yelp Review	англ.	Accuracy	67.8	**68.6**
Kaggle Insults	англ.	ROC-AUC	86.1	**88.4**
SST	англ.	Accuracy	64.4	**66.1**

Таблица 5 — Результаты обучения моделей классификации на основе моделей BERT разных доменов (языковых стилей) для русского и английского языков.

– Классификационные модели, обученные на векторных представлениях языковых моделей ELMo, достигают результатов, сравнимых с результатами моделей на основе `fastText`, при несоответствии домена языковой модели и целевой задачи.
– Классификационные модели, обученные на векторных представлениях языковых моделей ELMo соответствующего целевой задаче домена, значительно превосходят результаты как моделей, использующих `fastText` соответствующего домена, так и всех моделей, обученных на данных другого домена.
– На данных разговорного стиля векторные представления моделей `fastText`, ELMo и BERT соответствующего домена позволяют достичь лучших результатов при решении задачи классификации текстов как для английского, так и для русского языков.
– Представленные в данной главе модели классификации и векторные представления выложены в открытый доступ в библиотеке DeepPavlov[20].
– Результаты исследований, представленные в данной главе, опубликованы в работах [18—20], а также представлены в постере «Conversational BERT for English and Russian languages» на конференции «AI Journey».

Несмотря на то, что исследования, представленные в данной главе, проводились в 2017–2019 годы, сделанные выводы и предложенный подход к обучению моделей классификации разговорного домена на основе векторных представлений языковых моделей соответствующего домена являются актуальными и на данный момент. В диалоговой системе DREAM были встроены сразу

[20] http://docs.deeppavlov.ai/en/master/features/models/classifiers.html

несколько классификаторов, которые используют в качестве основы модель BERT разговорного домена.

Применение моделей классификации на базе предобученных языковых моделей разговорного стиля открытого домена в диалоговой системе DREAM представлено в Главе 3. Предложенные модели определения тональности и токсичности проанализированы в качестве автоматических метрик для демонстрации диалоговой системой здравого смысла в Главе 4. Предложенная модель определения токсичности, аналогично построенные модели предсказания вероятности прерывания диалога `Dialogue Termination` и предсказания вероятности несоответствия контексту `Dialogue Breakdown` (описанные в Главе 3) используются в алгоритме выбора финального ответа, представленного в Главе 5.

Глава 3. Диалоговая система DREAM

В данной главе будет описан конкурс «Alexa Prize Challenge» в Разделе 3.1, архитектура и основные компоненты диалоговой системы DREAM во время участия в конкурсах «Alexa Prize Challenge 3» в Разделе 3.2 и «Alexa Prize Challenge 4» в Разделе 3.3. В Разделе 3.4 подробно описаны несколько сценарных навыков, разработанных автором диссертации.

Автором были разработаны некоторые аннотаторы и навыки в диалоговой системе DREAM, остальные компоненты разработаны членами команды Московского физико-технического института. Фреймворк `DeepPavlov Agent` разработан коллегами по лаборатории нейронных систем и глубокого обучения.

3.1 Конкурс «Alexa Prize Challenge»

«Alexa Prize Socialbot Grand Challenge» от «Amazon» – это международный конкурс диалоговых систем открытого домена, которые могут общаться с пользователями колонок «Amazon Alexa» на различные популярные темы. Пользователи могут включить бота с помощью команды «Alexa, let's chat» («Алекса, давай поболтаем»), чтобы поговорить с ним. По окончании разговора пользователю будет предложено оценить, насколько он хотел бы поговорить с этим ботом еще раз по шкале от 1 до 5, поставить так называемый рейтинг (англ.: rating). Глобальная задача конкурса – создание диалоговой системы, среднее время разговора с которой составляет более 20 минут и средний рейтинг больше 4. Но в связи со сложностью глобальной задачи, конкурс является поэтапным, и каждый год «Amazon» проводит «Alexa Prize Challenge», в котором из большого числа заявок отбираются до 10 университетских команд для участия в конкурсе продолжительностью более полугода.

Конкурс состоит из нескольких этапов: (1) подготовительный период, (2) период бета-тестирования на пользователях-сотрудниках «Amazon», (3) период начальной обратной связи (англ.: Initial Feedback Period), когда боты-участники впервые становятся доступны обычным пользователям колонок , (4) четверть-финалы (англ.: Quarterfinals), по рейтингам за последнюю

неделю которых отбираются команды для прохода в полуфиналы, (5) полуфиналы (англ.: Semifinals), средний рейтинг за весь период которых учитывается при отборе команд для прохода в (6) финалы (англ.: Finals), во время которых диалоговые системы оцениваются специально обученным жюри для выбора итогового победителя.

Команда Московского физико-технического института DREAM была отобрана для участия в конкурсах «Alexa Prize Challenge 3»[1], в котором принимала участие автор[2], и «Alexa Prize Challenge 4»[3], в котором автор была капитаном команды[4]. В обоих конкурсах команда DREAM дошла до полуфиналов, но не была отобрана для прохода в финал. Диалоговая система DREAM [21; 22; 24] построена на базе библиотеки DeepPavlov Agent, которую разрабатывает лаборатория нейронных систем и глубокого обучения, на базе которой и была собрана команда для участия в конкурсе. Все другие команды использовали внутренний фреймворк CoBot от «Amazon».

3.2 Диалоговая система DREAM в конкурсе «Alexa Prize Challenge 3»

Диалоговая система DREAM для участия в конкурсе «Alexa Prize Challenge 3» строилась с нуля командой Московского физико-технического института на базе фреймворка DeepPavlov Agent, который разрабатывается непосредственно сотрудниками лаборатории нейронных систем и глубокого обучения МФТИ. Разработка DREAM началась летом 2019 года, и в связи с тем, что DeepPavlov Agent находился на ранней стадии разработки, было принято решение использовать фреймворк не как библиотеку фиксированной версии, а полностью скопировать программный код библиотеки в репозиторий с диалоговой системой. В дальнейшем это позволило оперативно вносить правки и интегрировать новые возможности, которые впоследствии были добавлены в DeepPavlov Agent. На Рисунке 3.1 представлена верхнеуровневая архитектура диалоговой системы DREAM на момент окончания конкурса «Alexa Prize

[1] https://developer.amazon.com/alexaprize/challenges/past-challenges/challenge3
[2] https://deeppavlov.ai/challenges/dream_alexa_3
[3] https://developer.amazon.com/alexaprize/challenges/current-challenge/teams
[4] https://deeppavlov.ai/challenges/dream_alexa_4

Challenge 3». Задача оркестрирования контейнерным кластером в диалоговой системе DREAM осуществлялась с помощью Docker Compose[5] в рамках конкурса «Alexa Prize Challenge 3».

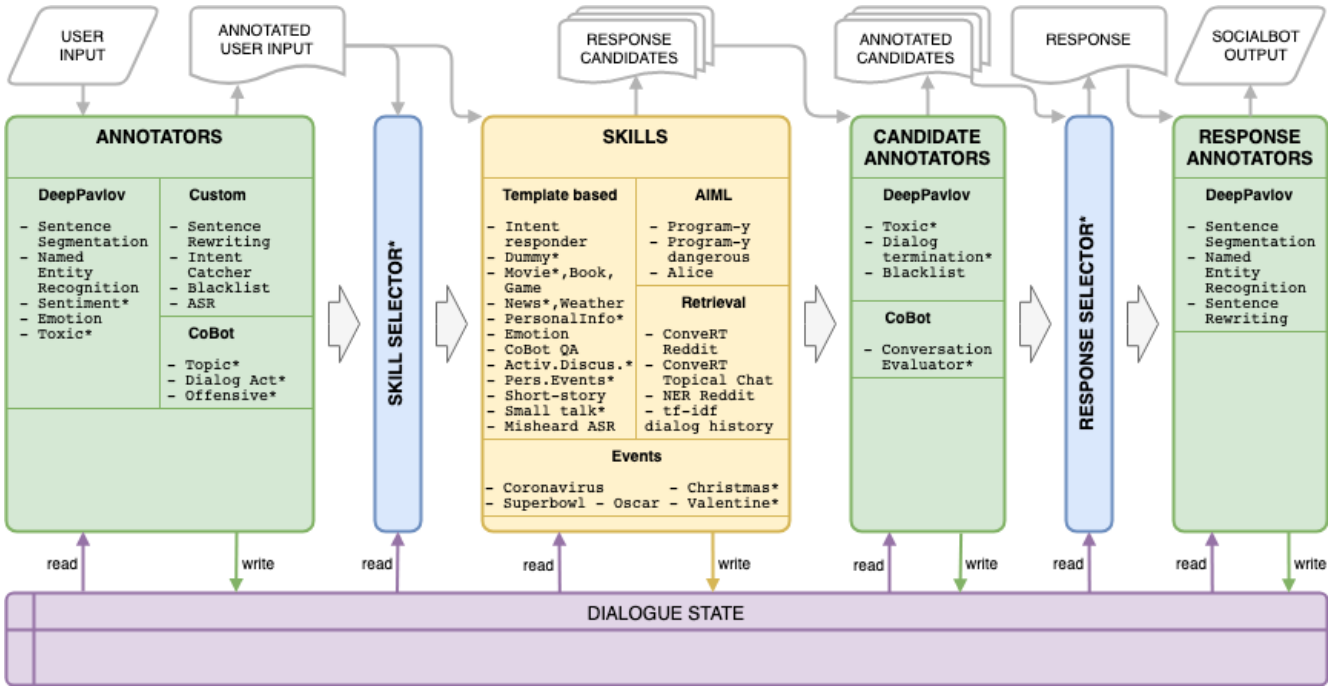

Рисунок 3.1 — Верхнеуровневая архитектура диалоговой системы DREAM в конкурсе «Alexa Prize Challenge 3». Символом «*» отмечены компоненты, преимущественно разработанные автором диссертации. Автор также принимала участие в разработке и правках других компонент.

Реплики пользователя поступают на вход агента в виде текстовой транскрипции голосового ввода. Например, если пользователь произнес фразу «Do I like unicorns?», на вход поступает список распознанных слов с соответствующими показателями уверенности модели распознавания речи: [(«do», 0.95), («i», 0.98), («like», 0.99), («unicorns», 0.85)]. Поэтому в первую очередь, для более корректной обработки реплик, были добавлены модели восстановления пунктуации и разделения реплик на предложения `Sentence Segmentation`, а также модель `Sentence Rewriting`, которая делает минимальное разрешение кореференции, заменяя местоимения на упомянутые ранее в диалоге сущности.

На начальном этапе разработки диалоговой системы были интегрированы предоставленные командой «Amazon» удаленные сервисы: классификатор тем `CoBot Topic Classifier`, комбинированный классификатор диалоговых актов и тем `CoBot DialogAct Classifier`, вопросно-ответная система `CoBotQA`, которая способна отвечать на фактоидные вопросы и некоторые популярные

[5] https://docs.docker.com/compose/

вопросы про персону «Alexa». Также оперативно были добавлены навыки открытого домена, такие как `Alice`, `AIML Chit-Chat`, которые подробно описаны в Разделе 3.4.

По мере необходимости добавлялись различные аннотаторы реплик пользователя, начиная с необходимых фильтров, таких как `Blacklist Words Detector` – основанный на словарях детектор «плохих» и «опасных» слов и выражений – и `Toxic Classifier` – нейросетевой классификатор, определяющий, содержит ли реплика оскорбления, угрозы, нецензурные или неприличные выражения, ненависть или иные проявления токсичности. Для создания аннотатора `Toxic Classifier` модель «conversational BERT», используемая для классификации разговорных текстов[6] и являющаяся частью фреймворка DeepPavlov, была дообучена на данных с «Kaggle Toxic Comment Classification Challenge»[7]. Также были интегрированы нейросетевые модели классификации тональности `Sentiment Classifier` и эмоций `Emotion Classifier`, которые также основаны на модели «conversational BERT»[6]. `Sentiment Classifier`[8] обучен на наборе данных Stanford Sentiment Treebank [82], а `Emotion Classifier`[9] на данных с Kaggle-страницы Eray Yildiz[10] (на момент написания диссертации данные убраны из открытого доступа) и ScenarioSA [83]. Модели используют подход, описанный в Главе 2 данной диссертации, основанный на использовании доменно-специфичных языковых моделей, в частности разговорной стилистики.

В связи с большим количеством правил и ограничений на поведение диалоговой системы от команды «Amazon», были изменены некоторые шаблоны и добавлены новые в используемых готовых навыках, а также разработан специальный навык `Intent Responder`, который включается при обнаружении аннотатором `Intent Catcher` определенных запросов, чтобы диалоговая система могла немедленно отреагировать и выдать специальные шаблонные ответы. Например, при запросе имени или команды-создателя бота, диалоговая система должна отказаться делиться этой информацией в связи с правилами конкурса. Компонента `Intent Catcher` использует набор регулярных выражений и клас-

[6] http://docs.deeppavlov.ai/en/master/features/pretrained_vectors.html#bert

[7] https://www.kaggle.com/c/jigsaw-toxic-comment-classification-challenge/overview

[8] https://github.com/deepmipt/DeepPavlov/blob/0.9.0/deeppavlov/configs/classifiers/sentiment_sst_conv_bert.json

[9] http://files.deeppavlov.ai/datasets/EmotionDataset.rar

[10] https://www.kaggle.com/eray1yildiz

сификатор на 21 класс, который основан на последней на момент проведения конкурса[11] версии Universal Sentence Encoder [84].

В выборщик навыков `Skill Selector` была заложена довольно простая логика, которая основана на постоянном включении навыков открытого домена, включении тематических навыков (закрытого домена) при определенных условиях на аннотации реплики пользователя, а также использовании безопасного режима для обработки токсичных реплик или реплик на спорные темы. Подробнее `Skill Selector` описан в Разделе 5.1. Выборщик ответов `Response Selector` на начальном этапе выбирал финальную реплику с максимальным значением показателя уверенности – вещественного показателя от 0 до 1, назначаемого навыком каждой возвращаемой реплике-кандидату. Далее был интегрирован удаленный сервис `CoBot Conversation Evaluator` [85] в качестве аннотатора реплик-кандидатов. Предсказания от данного сервиса также были добавлены в финальную схему работы `Response Selector` в рамках конкурса «Alexa Prize Challenge 3», которая подробно описана в Разделе 5.2.

Активная работа велась и над созданием навыков закрытого домена для поддержания разговоров на некоторые популярные темы и над созданием навыков открытого домена для обеспечения хотя бы минимального покрытия как можно большего числа тем. Был добавлен упомянутый ранее ранжирующий навык (ранжирующие навыки описаны в Разделе 1.4.1) `ConveRT Reddit`, использующий нейросетевую модель ConveRT [48] для получения векторных представлений реплики и контекста, который получается конкатенацией реплик из истории диалога. Модель ConveRT имеет меньшее число параметров и работает быстрее, чем модели векторных представлений на основе BERT, при этом имеет сопоставимое качество векторных представлений. Модель возвращает реплики-кандидаты, ранжируя пары контекст-реплика по косинусной близости соответствующих векторных представлений. Модель была обучена на большом наборе комментариев с сайта Reddit[12] для обеспечения разговорной стилистики. Около 2 миллионов комментариев были собраны с сайта Reddit и отфильтрованы сервисами CoBot `Conversation Evaluator` и `Toxic Classifier`. В результате в окончательном наборе реплик-кандидатов для ранжирования осталось 80 тысяч комментариев. Данный навык отлично проявил себя и использовался на протяжении обоих конкурсов. Одновременно были

[11] https://tfhub.dev/google/universal-sentence-encoder/4

[12] https://reddit.com

добавлены базовые навыки закрытого домена `TF-IDF-retrieval Skills on Topical Chat` – набор ранжирующих навыков, использующих векторные представления TF-IDF, обученные на наборах TopicalChat [6], PersonaChat [7] и Wizards-of-Wikipedia [8], и реплики из датасета Topical Chat [6]. Навыки покрывают различные популярные темы: книги, развлечения, мода, фильмы, музыка, политика, технологии, спорт и животные.

В процессе выборочной оценки диалогов с реальными пользователя стало понятно, что ранжирующих навыков закрытого домена для покрытия популярных тем не достаточно, в связи с чем были разработаны следующие сценарные тематические навыки (сценарные навыки описаны в Разделе 1.4.1): `Movie Skill`, `Book Skill`, `Game Skill`, `News Skill`, `Weather Skill`, `Emotion Skill`, `Personal Info Skill`, `Coronavirus Skill`, `Christmas Skill`, `Valentine's Day Skill`, `Superbowl Skill`, `Oscar Skill`. `Movie Skill` и `News Skill` для обсуждения фильмов и новостей подробно описаны в Разделе 3.4. Также разработанные автором сценарные навыки `Personal Info Skill`, `Christmas Skill`, `Valentine's Day Skill` используют набор шаблонов для извлечения и сохранения личной информации (имя, родной город, город проживания) о пользователе и для ведения связного диалога об известных событиях. Кроме того, автором были разработаны сценарные навыки, интегрирующие здравый смысл в диалог, которые подробно описаны в Главе 4. Автором также был разработан разговорный навык `Small Talk Skill` для проведения сценарных диалогов по большому числу различных тем. Основой ветвления сценариев явилось определение намерений согласия и несогласия пользователя на задаваемые системой вопросы для развития темы. В следующем конкурсе данный навык стал прообразом для создания большого числа сценарных навыков.

Появление сценарных навыков было направлено на решение проблемы развития диалога, однако, несмотря на то, что навыки обеспечивают связный диалог на несколько шагов на популярную тему, пользователю необходимо сначала попасть в данный навык. Соответственно, возникла необходимость в методике направления развития диалога с максимизацией использования сценарных навыков, причем переходы от одного навыка к другому должны быть плавными. Юсуповым Идрисом, членом команды DREAM и соавтором статьи [21], был предложен метод использования «направляющих вопросов» («linking questions», «link-to questions»), которые используются в диалоге для направления пользователя в обсуждение покрытых сценариями тем. Обычно

такие вопросы подразумевают в качестве ответа выражение мнения о заданной теме или выбор пользователем некоторого объекта для обсуждения, например, фильма, книги или игры. Данные вопросы могут добавляться к репликам как внутри самих сценарных навыков по окончании сценария, так и в `Response Selector`.

Ближе к концу конкурса, реплики-кандидаты также аннотировались с помощью `Dialogue Termination`, который предсказывает, собирается ли пользователь завершить диалог (например, сказать «Alexa, stop») на следующем шаге, то есть после выдачи данной реплики-кандидата в качестве финальной. Модель на основе «conversational BERT» от DeepPavlov[6] была обучена на данных, собранных за время разговоров диалоговой системы DREAM с пользователями «Alexa». `Response Selector` использовал данные аннотации, отфильтровывая реплики-кандидаты со значением вероятности завершения диалога выше порогового значения.

Базовый алгоритм выбора финального ответа, в разработке которого участвовала автор, подробно описан в Разделе 5.2. Верхнеуровневый алгоритм следующий: фильтруются «плохие реплики-кандидаты»; набором правил для приоритизации определенных навыков в некоторых случаях повышаются уровни уверенности навыков; подсчитывается финальное значение показателя оценки для каждой реплики-кандидата с помощью эмпирической формулы с учетом уверенности навыка и оценок от `CoBot Conversation Evaluator`; к финальному ответу в определенных случаях присоединяются «направляющий вопрос» и/или обращение по имени к пользователю.

3.3 Диалоговая система DREAM в конкурсе «Alexa Prize Challenge 4»

Диалоговая система DREAM для участия в конкурсе «Alexa Prize Challenge 4» была изначально построена на базе доработанной за промежуток между конкурсами финальной версии диалоговой системы DREAM после конкурса «Alexa Prize Challenge 3». На Рисунке 3.2 представлена верхнеуровневая архитектура диалоговой системы DREAM на момент окончания конкурса «Alexa Prize Challenge 4». В новой версии системы решались следующие задачи:

моделирование предпочтений пользователей, целеориентированное управление диалогом (подробнее в Главе 5) и масштабирование домена. Задача оркестрирования контейнерным кластером в диалоговой системе DREAM осуществлялась с помощью Kubernetes[13].

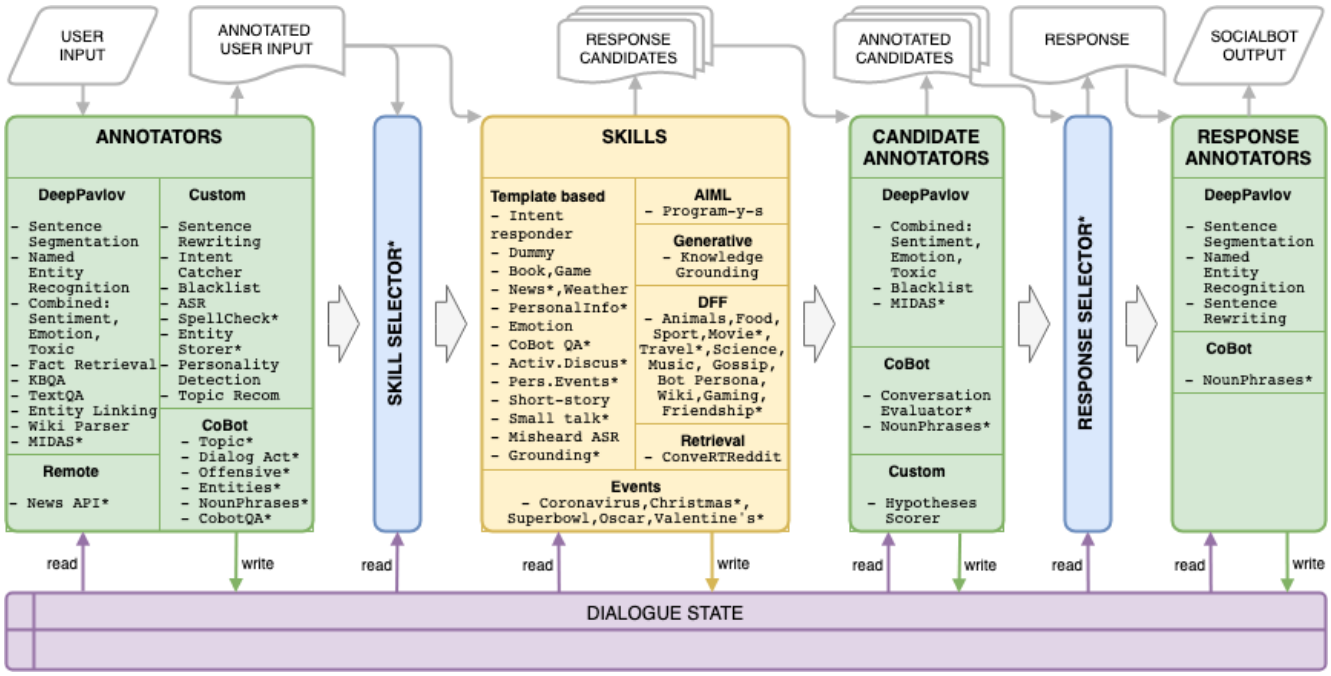

Рисунок 3.2 — Верхнеуровневая архитектура диалоговой системы DREAM в конкурсе «Alexa Prize Challenge 4». Символом «*» отмечены компоненты, преимущественно разработанные автором диссертации. Автор также принимала участие в разработке и правках других компонент.

В диалоговой системе DREAM в первую очередь были сделаны изменения, касающиеся оптимизации потребляемых ресурсов и замещения использованных в «Alexa Prize Challenge 3» удаленных сервисов. Во-первых, аннотации `CoBot Topic Classifier` и `CoBot DialogAct Classifier` реальных диалогов с пользователями, полученных за все время конкурса «Alexa Prize Challenge 3», были собраны в большой набор данных для классификации реплик по темам и диалоговым актам. В итоге, были объединены 6 моделей-классификаторов `Emotion Classifier`, `Sentiment Classifier`, `Toxic Classifier`, `CoBot Topic Classifier` и `CoBot DialogAct Classifier` (возвращает две метки: тему и диалоговый акт). Подробная информация о наборах данных и результатах обучения представлена в [24]. Также был добавлен новый классификатор диалоговых актов `MIDAS Classifier` на базе датасета MIDAS [86], из которого были использованы метки семантических классов. Данная компонента также

[13] https://kubernetes.io

основана на модели «conversational BERT», используемой для классификации разговорных текстов[14]. Модель не выложена в открытый доступ, так как датасет на данный момент также убран из открытого доступа и был получен командой напрямую от создателей датасета. Также был внедрен аннотатор `CoBot Entities`, который стал доступен участникам как удаленный сервис в середине конкурса и был предназначен для извлечения сущностей и их классификации на несколько видов, например, «person», «videoname», «sport», «misc» и другие.

Одной из основных целей разработки диалоговой системы во время конкурса стала интеграция баз знаний. Так, например, были добавлены компоненты `Entity Linking`, `Wiki Parser` и `Fact Retrieval`. `Entity Linking` каждой сущности, распознанной с помощью `CoBot Entities`, сопоставляет ее идентификатор в системе Wikidata[15]. Далее полученные идентификаторы передаются в `Wiki Parser`, который извлекает из графа знаний Wikidata KG (Knowledge Graph) триплеты. *Триплет* в графах знаний – это набор *(субъект, отношение, объект)*, представляющий собой некое знание о сущности-субъекте, которое может быть переписано в текстовый факт с использованием специальных шаблонов. Новый компонент `Fact Retrieval` для сущностей, извлеченных с помощью `CoBot Entities`, получает факты из Wikipedia и wikiHow[16], где факты представляют из себя короткие параграфы текста из специально отобранных для каждого типа сущности списка подзаголовков статьи сайта Wikipedia или wikiHow. Также из всех навыков, которые получали информацию относительно реплики пользователя из интернета, были извлечены модули запросов к удаленным сервисам и выделены в отдельные аннотаторы, например, `CoBotQA Annotator`, `News API Annotator`. Это позволило дать доступ к извлеченной информации всем навыкам.

В конкурсе «Alexa Prize Challenge 3» победила команда, имплементировавшая большое количество сценариев в диалоге, покрывающих большинство популярных тем и обеспечивающих связный диалог «в глубину» минимум на несколько шагов. В диалоговой системе DREAM на момент окончания конкурса «Alexa Prize Challenge 3» было недостаточное количество сценарных навыков для покрытия всех тем. Существующие сценарные навыки было невоз-

[14] http://docs.deeppavlov.ai/en/master/features/pretrained_vectors.html#bert

[15] http://wikidata.org/

[16] https://www.wikihow.com/Main-Page

можно универсально распространить на другие темы, за счет чего возникали сложности с разработкой сценарных навыков. Поэтому в рамках «Alexa Prize Challenge 4» был предложен и разработан `Dialogue Flow Framework` (`DFF`)[17] – специальный фреймворк для построения диалоговых систем, позволяющий удобно записывать сценарий в виде диалогового графа, вершины и ребра которого представляют из себя специальные функции проверок и генерации ответа на языке `Python`. Сразу после выпуска первой версии данного фреймворка была начата разработка сценарных тематических навыков: `Animals Skill`, `Food Skill`, `Sport Skill`, `Science Skill`, `Music Skill`, `Gossip Skill`, `Gaming Skill`, `Bot Persona Skill`, `Travel Skill`, а также переведен на использование `DFF` навык `Movie Skill`. Последние два созданы автором и подробно описаны в Разделе 3.4. Сценарные навыки покрывают большой список популярных тем, создавая впечатление связного диалога «в глубину» выбранной темы. Благодаря введению данных сценарных навыков было прекращено использование тематических ранжирующих навыков, упомянутых в Разделе 3.2, а также навыка `Small Talk Skill`, разработанного автором.

Важными выводами из использования сценарных навыков являются ограниченность лексического разнообразия и невозможность покрыть все популярные темы и их подтемы. Поэтому была необходима разработка универсальных сценарных навыков открытого домена, то есть способных обсуждать произвольной объект на базе некоторого сценария. Для этого был разработан навык `Wiki Skill`, который изначально был предназначен для того, чтобы вести диалог о конкретном объекте, понятии или явлении с использованием структуры соответствующей страницы сайта Wikipedia, то есть обеспечить структурированный (фактически, сценарный) диалог по большому количеству объектов. В дальнейшем, вдохновляясь концепцией разговорного навыка `Small Talk Skill`, `Wiki Skill` был расширен до возможности строить сценарный диалог, в котором есть возможность добавлять обсуждение страниц сайта Wikipedia, в том числе об извлеченных из реплики пользователя сущностях. Задача `Wiki Skill` в том, чтобы поддерживать сценарный диалог по большому списку популярных объектов, не охваченных отдельными сценарными навыками.

Во время конкурса «Alexa Prize Challenge 3» командой DREAM была сделана попытка внедрения генеративных навыков (генеративные навыки

[17] https://github.com/deepmipt/dialog_flow_framework

описаны в Разделе 1.4.1), однако они не продемонстрировали результата, достаточного для использования на реальных пользователях. Поэтому одним из важных направлений разработки стало внедрение генеративных моделей в диалоговую систему. Для большего контроля генерируемых реплик, в навыке `Knowledge Grounding Skill` была использована нейросетевая генеративная модель ParlAI Blender 90M [50], дообученная на наборе диалогов Topical Chat Enriched [51].

В [21; 22] командой DREAM в рамках конкурса «Alexa Prize Challenge 3» был представлен подход к связыванию разговорных навыков в диалоге с помощью «направляющих вопросов». Для конкурса «Alexa Prize Challenge 4» был внедрена новая компонента `Topic Recommendation`, которая на основе текущего контекста дает рекомендацию, какая следующая тема может быть использована. На основе предсказаний от `Topic Recommendation` выбираются направляющие вопросы, которые ведут пользователя в сценарный навык, поддерживающий рекомендованную тему. Подробное описание компоненты `Topic Recommendation` и экспериментов с ней представлено в [24].

В связи с наличием большого числа сценарных навыков на различные популярные темы, важной задачей `Response Selector` также становится обеспечение плавного перехода к следующей теме. Переход между некоторыми навыками сделан с помощью узко-специфичных вопросов, которые плавно переводят пользователя от одной темы к другой, а значит переводят управление диалогом от одного сценарного навыка к другому. Например, в рамках обсуждения города проживания пользователя бот предложит узнать прогноз погоды в этом городе, или в конце обсуждения традиционной еды различных стран бот спрашивает, посещал ли пользователь эту страну и переводит разговор на тему путешествий. Однако обеспечить такой переход между всеми навыками не представляется возможным, так как подразумевает $N * (N - 1)$ переходов, где N – число навыков.

В случае отсутствия обоснованного перехода с одной темы на другую может возникнуть впечатление несвязности диалога. Поэтому почти для всех пар тем был собран набор цитат, интересных фактов и идей, относящихся к обеим темам одновременно. В случае, если для текущей пары тем (тема, по которой велось обсуждение со сценарными навыком в последних 5 репликах, и тема, на которую должен быть сделан переход) отсутствует специфичный переход, перед направляющим вопросом добавляется «связующая фраза». Тем самым

создается впечатление связности и плавности перехода от одной темы к другой. Например, для перехода от `Movie Skill` к `Travel Skill` (и наоборот) может использоваться следующая комбинация:

«More and more travelers are visiting locations after seeing them featured in a movie, like Joker's stairs in the Bronx. What place do you want to visit someday?»

Сильно увеличивающееся разнообразие количества и происхождения навыков привело к необходимости изменить алгоритм выбора финального ответа для приоритизации сценарных навыков. Стратегия приоритизации сценарных навыков основана на том, что проведение пользователя по сценарию создает положительное впечатление, а также позволяет лучше ориентироваться в диалоге самому боту. Алгоритм выбора финального ответа «Response Selector» предложен автором и подробно описан в Разделе 5.4.

3.4 Примеры сценарных разговорных навыков

На данный момент именно сценарные разговорные навыки являются движущей силой диалога: они создают впечатление связного диалога в глубину на несколько ходов, раскрывают персону бота, извлекают персону (предпочтения, личную информацию) пользователя. В данном разделе описаны примеры разговорных навыков, использованных в диалоговой системе DREAM, построенных с использованием различных инструментов.

AIML Chit-Chat основан на фреймворке Program Y[18] и использует шаблонного бота[19], который внедрен в диалоговую систему как разговорный навык. Командой DREAM для участия в конкурсах «Alexa Prize Challenge 3» и «Alexa Prize Challenge 4» шаблоны были доработаны для приведения в соответствие с правилами конкурса. Например, навык отказывается давать медицинские и финансовые советы, не называет своих персональных данных, просит пользователя не употреблять ненормативную лексику и не оскорблять бота, при запросе предлагает темы для разговора в соответствии с умениями DREAM. Данный навык был ограничен довольно строгими шаблонами для реплики пользователя,

[18] github.com/keiffster/program-y

[19] github.com/keiffster/program-y/wiki/Available-Bots

для более полноценного использования Program Y был также внедрен аналогичный навык `AIML Chit-Chat General`, который использовал в том числе более общие шаблоны для реплики пользователя, а значит покрывал универсальными ответами большее число ситуаций, но отвечал с меньшим значением показателя уверенности. Навык `AIML Chit-Chat` можно отнести к сценарным навыкам, так как AIML позволяет осуществлять проверку нескольких последовательных реплик пользователя в истории и, соответственно, создавать небольшие сценарии в навыке. Аналогичным образом к сценарным относится и следующий навык `Alice`.

Alice – это AIML бот с открытым исходным кодом[20]. Он содержит исчерпывающий набор шаблонов, включая обсуждения различных популярных тем. Данный навык был особенно полезен в начале соревнования «Alexa Prize Challenge 3» в виду своей универсальности. В середине конкурса «Alexa Prize Challenge 4» данный навык был отключен.

CoBotQA – это специальный навык, интегрирующий использование удаленного сервиса Q&A CoBot. Навык может ответить на некоторые вопросы о личности бота и большинство фактоидных вопросов, а также используется для получения фактов об извлеченных из реплики пользователя сущностях с помощью запросов «fact about ENTITY» («факт о СУЩНОСТИ»). Удаленный сервис Q&A CoBot принимает на вход обычный текстовый запрос, точный алгоритм работы сервиса не доступен, однако известно, что сервис использует базу знаний Amazon Evi[21] для ответа на фактоидные вопросы. Вывод сервиса Q&A CoBot ограничивается 1-2 предложениями, дополняется вступительным фразами и иногда завершающими запросами мнения, что сделано для имитации разговорного стиля. Сам сервис Q&A CoBot также может давать разговорные ответы на некоторые виды запросов. Так как сам по себе навык является одношаговым, то не относится к сценарным навыкам, однако навык `CoBotQA` был расширен до последовательной выдачи фактов о последней упомянутой сущности в предыдущем факте, если пользователь заинтересован в продолжении.

[20] github.com/sld/convai-bot-1337/tree/master/ALICEChatAPI
[21] https://www.evi.com/

Movie Skill предназначен для обсуждения с пользователем фильмов. Изначально навык был построен без использования дополнительных инструментов, однако после выпуска `Dialog Flow Framework` (`DFF`) навык был переведен на его использование. Навык использует данные IMDb[22] в качестве базы данных. При запросе разговора о фильмах, кино, сериалах, мультфильмах и прочих релевантных объектах `Movie Skill` начинает свой сценарий, задавая случайный вопрос, подразумевающий в ответе пользователя название фильма (например, вопросы «What is your favourite movie?», «What would you recommend me to watch?» и другие). Также случайный вопрос может быть задан самой диалоговой системой для направления пользователя в сценарий про фильмы, если пользователь не сказал желаемую тему (подробнее про направляющие вопросы в Разделе 3.2). Далее навык ведет диалог по сценарию, сконцентрированному на определенном фильме, извлеченном из высказывания пользователя. Если название фильма не обнаружено в реплике пользователя, то навык предлагает пользователю рекомендации фильмов, которые основаны на вручную отобранном списке непопулярных (больше 1 тысячи и меньше 10 тысяч голосов), но имеющих высокий рейтинг (больше 8 из 10) фильмов на IMDb. Если же `Movie Skill` обнаруживает непопулярный по количеству голосов (меньше 10 тысяч голосов) на IMDb фильм, навык уточняет, является ли извлеченное название правильно распознанным. Если фильм популярный или название уточнено, то далее сценарий диалога о фильме включает шаблонные обмен мнениями, вопросы о жанре фильма, актерах или персонажах, а также выдачу интересных фактов об обсуждаемом фильме, получаемых с помощью `CoBotQA`. Навык `Movie Skill` также имеет возможность при завершении диалога о конкретном фильме предлагать рекомендации фильмов пользователю. В рамках навыка `Movie Skill` также была сделана попытка интегрировать использование нейросетевой генеративной модели `Knowledge Grounding Service`, которая обуславливается на заданное знание. Для этого был добавлен шаг сценария, на котором диалоговая система шлет запрос «What is your favourite moment?» в `Knowledge Grounding Service` с описанием сюжета фильма в качестве дополнительной информации. Полученный ответ от генеративной модели дополняется комментарием о том, что он иллюстрирует особо запомнившийся для бота момент фильма и вопросом про аналогичный момент для пользователя. К сожалению,

[22] https://www.imdb.com/interfaces/

генеративная модель зачастую плохо генерировала ответ на такой контекст, поэтому временно этот ход сценария был отключен.

Travel Skill предназначен для обсуждения с пользователем путешествий. Навык построен на базе `DFF` фреймворка. При запросе поговорить о путешествиях, `Travel Skill` начинает свой сценарий, задавая случайный вопрос, подразумевающий в ответе пользователя название локации (например, вопросы «What country would you like to visit?», «What is the most interesting place you have vsitied?» и другие). Аналогично `Movie Skill`, случайный вопрос может быть задан как направляющий вопрос (подробнее в Разделе 3.2). Далее навык ведет диалог по сценарию, сконцентрированному на определенной локации, извлеченной из реплики пользователя. Если название локации не найдено аннотаторами в реплике пользователя, то навык предлагает пользователю обсудить одно из нескольких популярных направлений для путешествий. Диалог о локации состоит в обсуждении, посещал ли пользователь это локацию, что больше всего его впечатлило, а также выдачу фактов о локации, получаемых с помощью `CoBotQA`, а также автоматически скачанных с нескольких сайтов с популярными фактами.

News Skill делится мировыми новостями с пользователем. В качестве источника новостей используется сервис GNews API[23], для которого в рамках конкурса была приобретена подписка для обеспечения большого числа запросов. Сервис также предоставляет бесплатное API с ограниченным числом запросов. Одноименный аннотатор для каждой сущности в реплике пользователя делает запросы к сервису и сохраняет в аннотации новости о данной сущности, а также для каждой реплики в аннотации сохраняется последняя актуальная новость. Новости кэшируются внутри аннотатора и обновляются с заданной периодичностью. При обнаружении непосредственного запроса новостей в реплике пользователя на определенную тему, для нее в аннотациях уже хранятся новости, так как тема была упомянута в реплике. Сценарий состоит в выдаче заголовка новости с предложением узнать подробности, если пользователь соглашается, навык также выдает первые несколько предложений содержания новости и спрашивает у пользователя его мнение. На следующем

[23] https://gnews.io

шаге `News Skill` предлагает на выбор темы новостей для дальнейшего обсуждения, если пользователь будет заинтересован. Во всех случаях, когда в реплике пользователя отсутствует прямой запрос новостей, но новости для упомянутых сущностей были найдены и сохранены в аннотациях, то навык возвращает реплику-кандидата с предложением узнать новости о заданной сущности.

По результатам, полученным в этой главе, можно сделать следующие **выводы**:

- Автор принимала непосредственное участие в разработке диалоговой системы DREAM, участвующей в международных конкурсах «Alexa Prize Challenge 3» и «Alexa Prize Challenge 4» англоязычных диалоговых систем открытого домена и прошедшей в полуфиналы обоих конкурсов.
- Для диалоговой системы DREAM автором были реализованы и внедрены для применения на реальных пользователях описанные в Главе 2 модели классификации на базе предобученных языковых моделей разговорного стиля.
- В данной главе предложены сценарные разговорные навыки, интегрирующие использование удаленных сервисов в качестве баз знаний, а также источника фактов.
- На разработанный автором разговорный навык `Movie Skill` оформлено свидетельство о государственной регистрации программы для ЭВМ № 2021664221 «Программа разговорного навыка для проведения диалога о кино» [25].
- На фреймворк `Dialog Flow Framework`, в разработке которого принимала участие автор, оформлено свидетельство о государственной регистрации программы для ЭВМ № 2021664168 «Среда для создания сценарных разговорных агентов» [28].
- Программный код последней версии диалоговой системы DREAM выложен в открытый доступ[24], в том числе выложены сценарные разговорные навыки, представленные в данной главе.
- Технические описания двух версий диалоговой системы DREAM, представленных в данной главе, опубликованы в работах [21], [22] и [24].

Сравнительный анализ демонстрации здравого смысла в диалоге с предложенными сценарными разговорными навыками представлен в Главе 4.

[24] https://github.com/deepmipt/dream

Подробное описание и анализ алгоритма диалогового менеджмента, предложенного автором, представлен в Главе 5.

Глава 4. Здравый смысл в диалогах

Одной из основных сложностей разработки разговорного искусственного интеллекта является проблема внедрения здравого смысла в диалог. Как было упомянуто в предыдущих главах, на данный момент широко распространены модульные диалоговые системы, сочетающие в себе использование *шаблонных*, *генеративных* (использующих нейросетевые модели предсказания последовательности слов или токенов) и *ранжирующих* (модели, выбирающие реплику по контексту среди заданных возможных реплик) навыков. При этом ни один из трех основных видов разговорных навыков не гарантирует использование или проявление здравого смысла ботом.

Разговорные навыки, основанные на шаблонах, обладают здравым смыслом, заложенным разработчиком, однако они сильно ограничены теми ситуациями, которые покрывают эти шаблоны. С одной стороны, шаблонные навыки явно решают задачу связности реплики бота и контекста, то есть проявляют заложенный разработчиком здравый смысл в контексте, но при этом все равно ограничиваются подмножеством рассматриваемых контекстов. С другой стороны, практически любой диалог на общие темы регулярно выходит за рамки, заложенные разработчиком. В таких ситуациях шаблонные навыки проявляют себя не лучшим образом, а за счет контраста проявляемой ботом разумности в рамках сценариев и неразумности в других случаях, у пользователя может складываться впечатление неестественности диалога или отсутствия здравого смысла у бота.

Несмотря на то, что современные методы использования предобученных языковых моделей [3—5] значительно улучшили качество решения многих задач обработки естественного языка, многие системы до сих пор плохо справляются с демонстрацией здравого смысла, что было показано в работах [14; 15; 87]. В отличие от шаблонных навыков, генеративные и ранжирующие разговорные навыки сами по себе зачастую производят осмысленные реплики, но при этом они не могут гарантировать смысловую связность в контексте.

На текущий момент существует уже несколько задач и датасетов для решения задач, связанных со здравым смыслом, такие как WinoGrande [88] и ART [89]. Эти задачи хорошо решаются человеком (94% для WinoGrande и

91.4% для ART), но по-прежнему являются сложными даже для современных предобученных языковых моделей.

Данный раздел посвящен исследованию интеграции здравого смысла в диалог с реальными пользователями. И хотя на данный момент не существует строгого и общепринятого определения *здравого смысла*, будем полагаться на следующие характеристики, введенные в [90]:

- общепринятость – здравым смыслом обладают и передают между собой люди;
- фундаментальность – утверждения, соответствующие здравому смыслу, принимаются как должное;
- неявность – люди обычно не формулируют полностью утверждения, соответствующие здравому смыслу, так как другие люди также обладают этим знанием, и в большинстве случаев короткой отсылки достаточно для взаимопонимания;
- масштабность – количество и разнообразие знаний, соответствующих здравому смыслу, неисчислимо;
- всеобъемлемость – здравый смысл покрывает все аспекты жизни человека, а не специфичный домен;
- принятие по умолчанию – здравый смысл представляет собой предположения по умолчанию о типичных случаях в повседневной жизни, поэтому большая часть этих предположений скорее правдоподобна, чем определенно верна.

Однако заданное выше определение здравого смысла более характерно для разговоров между людьми, и *человеческое восприятие здравого смысла собеседника в диалоге с ботом значительно отличается*. Поэтому в работе мы будем называть здравым смыслом определенный набор аспектов событий, которые могут быть предсказаны с помощью нейросетевой модели COMeT [91]: (1) об объекте действия: что чувствует объект – xAttr, какие последствия для объекта – xEffect, какие были причины у объекта – xIntent, что было необходимо объекту до – xNeed, реакция объекта – xReact, что может хотеть сделать объект после – xWant; (2) об окружающих: какие последствия для окружающих – oEffect, реакция окружающих – oReact, что могут хотеть сделать окружающие после – oWant. Также пользователи часто сомневаются в том, что бот может обладать здравом смыслом, и пытаются найти доказательства этому во время разговора. Поэтому, несмотря на задаваемое свойство *неявности*, в данной ра-

боте выделим *явную демонстрацию здравого смысла* как прямое использование знания в репликах (например, выражения причинно-следственных связей или использование свойств объекта). Более подробная информация о предлагаемом определении здравого смысла и разметке данных приведена в разделе 4.2.

В первую очередь, понимание ботом здравого смысла позволяет лучше понимать, о чем говорит пользователь: различные аспекты здравого смысла, характеризующие обсуждаемые объекты и действия, помогают лучше определить тему разговора, тональность, предположить последствия выдачи реплики. Это позволяет лучше контролировать ход диалога, в том числе делать более логичные переходы с темы на тему. С другом стороны более явное проявление здравого смысла ботом позволит улучшить пользовательский опыт. Использование знаний, соответствующих здравому смыслу, позволяет использовать более высокий уровень абстракций в диалоге. В разделе 4.1 будет также описано, как с помощью здравого смысла можно формировать собственное мнение бота.

В Разделе 4.1 подробно описаны оригинальные разговорные навыки, которые направлены на то, чтобы в максимальной степени продемонстрировать собеседнику обладание бота здравым смыслом. Подробная информация о рассматриваемом определении здравого смысла и разметке данных приведена в Разделе 4.2. В последнем Разделе 4.3 предложенные разговорные навыки сравниваются с несколькими разговорными навыками из диалоговой системы DREAM, описанными в Разделе 3.4 по распределению размеченного здравого смысла, а также исследуется корреляция автоматических метрик и здравого смысла. В данной работе используются модели предсказания здравого смысла COMeT [91] на графах знаний ATOMIC [92] и ConceptNet [93].

Исследования и результаты, представленные в данной главе, опубликованы в [23].

Автором были разработаны разговорные навыки, интегрирующие модели предсказания здравого смысла, предложенные в данной главе. Методология разметки, разметка, и анализ корреляции были проработаны и осуществлены совместно с соавторами работы [23].

4.1 Разговорные навыки, интегрирующие здравый смысл

Повседневные разговоры на общие темы часто содержат упоминания и обсуждения различных занятий. Некоторые виды деятельности можно обсуждать, сосредоточившись на субъекте действия, например, «играть на фортепиано», «изучать географию» можно свести к обсуждению «фортепиано» и «географии» соответственно. Однако некоторые занятия не включают в себя субъекты (например, «купаться», «устать»), а другие формируются совместным смысловым вкладом действия и субъекта (например, «дрессировать собаку», «смотреть кино»). В связи с этим возникает необходимость разработки разговорных навыков, обладающих следующими свойствами: (1) способных поддерживать разговор о самых разных сферах человеческой деятельности; (2) способных говорить о человеческой деятельности не только на основе информации из графов знаний, но также с точки зрения чувств, мотивации, последствий. Все эти аспекты можно обобщить как проявление здравого смысла, и, благодаря нейросетевым моделям предсказания здравого смысла COMeT [91], они могут быть извлечены для любых выражений действия.

Для обоих предлагаемых навыков обсуждаемые занятия извлекаются из реплики пользователя с использованием оригинального метода, включающего морфологический анализ, извлечение сущностей, использование частот слов и выражений. Сущности для выражения личного мнения бота извлекаются с помощью морфологического анализа или модулей распознавания именованных сущностей [21].

4.1.1 Activity Discussion Skill

`Activity Discussion Skill` выражает *мотивацию бота лучше понимать человеческий мир.* Данный навык запрашивает у пользователя объяснение занятий человека, которые либо извлекаются из высказываний пользователя, либо случайным образом выбираются из заранее подготовленного списка увлечений. Навык делает предположения о различных характеристиках и причинно-следственных связях относительно обсуждаемого занятия, что

демонстрирует понимание ботом здравого смысла, а то, что реплики сформулированы в формате предположений, сглаживает возможные ошибки. Сам сценарий обсуждения занятий состоит из запроса на объяснение, нескольких уточняющих вопросов и запроса мнения пользователя. Навык использует шаблоны реплик со слотами *причинно-следственных связей* и *понимания человеческих эмоций*, которые заполняются с помощью предсказаний модели COMeT ATOMIC [91]. Например, для действия *«practice yoga»* модель COMeT ATOMIC генерирует *«be healthy»* как аспект намерения «xIntent». Следовательно, навык может сформировать ответ *«Do humans (practice yoga) [to be healthy]?»*, заполняя ячейки названием действия и предсказанным аспектом. Пример полного диалога с навыком представлен на Рисунке 4.1.

Human: I've *jumped with a parachute*!
Bot: What does *jumping with a parachute* mean? [explanation request]
Human: Yeah. It's a kind of risky activity.
Bot: Is it feels [scared] (xAttr) to *jump with a parachute*? [clarification]
Human: Definitely.
Bot: Why someone would need [to buy a parachute] (xNeed)? [clarification]
Human: No, you may rent it.
Bot: Great! Thank you. So, did you like *jumping with a parachute*?

Рисунок 4.1 — Пример диалога с `Activity Discussion Skill`. Не является диалогом с реальным пользователем в соответствии с правилами конкурса «Alexa Prize Challenge».

4.1.2 Personal Event Discussion Skill

`Personal Event Discussion Skill` имеет два режима: (1) выражение мнения о различных объектах, (2) обсуждение действий пользователя, чтобы продемонстрировать способность отслеживать причинно-следственные связи и устанавливать эмоциональный контакт с пользователем.

Первый режим `Personal Event Discussion Skill` предназначен для выражения обоснованного мнения о различных объектах и событиях в зависимости от тональности аспекта здравого смысла, предсказанного с помощью

COMeT ConceptNet [91]. Например, для запрошенного мнения о *«cats»* навык случайно выбирает аспект «SymbolOf», заполняет слот в ответе на основе шаблона предсказанием модели COMeT ConceptNet *«love»*, определяет позитивную тональность получившегося предсказания и формирует следующую реплику: *«I love (cats) so much because (cat) is a sign of [love].»*.

Второй режим позволяет реагировать на действия пользователя с точки зрения намерений, чувств, эффектов и последствий с помощью COMeT ATOMIC [91]. Например, если пользователь говорит *«I will go to the theater next weekend.»*, навык извлекает действие *«go to the theater»* и с учетом будущего времени формирует реплику следующим вопросом *«Did you [buy the tickets]?»*, заполняя шаблон предсказанием модели COMeT ATOMIC в аспекте «xNeed». Пример диалога с навыком представлен на Рисунке 4.2.

Human: Do you like *musicals*?
Bot: Well, I like *musicals*. Makes me want [to learn to sing] (CausesDesire). [opinion]
Human: Cool. I *went to the cinema* to see it on weekends.
Bot: Well. Did you want [to be entertained] (xWant) when *went to the cinema*? [clarification]
Human: Kind of.
Bot: So, hope you feel quite [excited] (xReact) now. [comment]
Human: Aha. Thanks.

Рисунок 4.2 — Пример диалога с `Personal Event Discussion Skill`. Не является диалогом с реальным пользователем в соответствии с правилами конкурса «Alexa Prize Challenge».

4.2 Разметка здравого смысла в диалогах

В данном разделе представлена схема разметки здравого смысла в диалоге. Как было описано ранее в начале Главы 4, здравый смысл характеризуется в том числе свойством неявности, так как люди обычно не произносят или пишут полностью утверждения, содержащие здравый смысл. Более того, здравый смысл используется в разговорах между людьми как общепринятые предполо-

жения о типичных ситуациях в повседневной жизни. В связи с этим необходимо выделить различные виды использования здравого смысла в диалоге.

Разговорные навыки бота могут генерировать ответы, которые демонстрируют или полагаются на здравый смысл, но при этом не соответствуют контексту. В этом случае, с точки зрения собеседника, бот либо противоречит здравому смыслу, либо дает бессмысленный ответ. Поэтому выделим *фразовый* (англ.: phrase-level) – независимый от контекста диалога – и *контекстный уровень* (англ.: context-level) здравого смысла. Для обоих уровней *фразового* и *контекстного* выделим выражающие *явный здравый смысл* (англ.: *explicit commonsense*), *неявный здравый смысл* (англ.: *implicit commonsense*), *бессмысленные* (англ.: *no sense*) и *неопределенные* (англ.: *undefined*) случаи.

Высказывание выражает *explicit commonsense*, если оно *явно* содержит утверждения, которые соответствуют здравому смыслу или предположения по умолчанию о типичных случаях повседневной жизни. Например, *«It's rainy outside, don't forget an umbrella»*, *«It feels so magical to see unicorns in a dream»* или *«You can pet a cat»*. Во всех этих случаях предположения по умолчанию явно указаны в высказывании. Значимые утверждения, которые явно не содержат утверждения здравого смысла, но ссылаются на них, определим как *implicit commonsense*. Например, фраза *«I like unicorns»* не содержит предположения по умолчанию (например, *«unicorns don't exist»*), но имеет смысл для обоих собеседников из-за общих знаний, соответствующих здравому смыслу (например, концепция единорогов, которые существуют только в сказочном мире). Класс *no sense* включает бессмысленные ответы и фразы, противоречащие здравому смыслу. Например, *«I like braavawqera»* не имеет смысла, в то время как фраза явно указывает отношение собеседника к «braavawqera», тем не менее, «braavawqera» не является частью контекста диалога или здравого смысла. В случаях, когда невозможно определить, требуется ли здравый смысл для ответа, назначается класс *undefined*.

Ответ бота считается выражением *explicit commonsense in a context*, если он соответствует контексту, и содержит здравый смысл на уровне фразы или дополняет контекст, явно выражая здравый смысл. Следовательно, для контекста *«What do you think about unicorns?»* оба ответа *«They are unreal»* и *«Unicorns are unreal»* выражают явный здравый смысл в контексте. В то время как последнее высказывание выражает явный здравый смысл в том числе на уровне фразы, высказывание *«They are unreal»* относится к неявному здраво-

му смыслу на уровне фразы, но с учетом контекста и того факта, что «they» подразумевает «unicorns», этот случай является демонстрацией явного здравого смысла в контексте. Другими примерами, иллюстрирующими случай, когда высказывание бота дополняет контекст до явного здравого смысла, являются следующие *«What is the color of the sky?» – «It's blue»* и *«I studied history in college» – «You have to be very smart»*.

Implicit commonsense in context соответствует реплике, которая уместна в данном контексте, но не включает явные утверждения здравого смысла или причинно-следственные связи, например *«What do you think about unicorns?» – «I like them»*. Если ответ не соответствует контексту или противоречит здравому смыслу на уровне фразы или в контексте, считаем его *no sense in context*. Класс *undefined in context* соответствует случаям, когда даже контекст не может помочь понять, был ли ответ осмысленным или нет.

Собрано по 100 примеров для каждого из 5 навыков из Раздела 3.4 и обоих предложенных навыков из Раздела 4.1. Три эксперта аннотировали каждый пример 2 метками: одна для фразового и одна для контекстного уровня здравого смысла. Итоговый датасет представляет собой комбинированный набор этих аннотаций, $100 \times 7 \times 3$ контекстов, каждый из которых имеет 2 метки (всего 4200 аннотированных примеров). Показатель согласованности между экспертами Каппа имеет значение 0.414, что интерпретируется как достаточное значение для честного сравнения.

4.3 Корреляция здравого смысла и автоматических метрик

4.3.1 Автоматические метрики

Предположим, что следующие автоматические метрики могут быть полезны для обнаружения здравого смысла: тональность и токсичность ответа пользователя на реплику бота, метрики оценки разговора и предсказания моделей логического вывода на основе ответа бота.

Классификатор тональности `Sentiment Classifier` определяет несет ли реплика позитивный, негативный или нейтральный характер. Нейросетевой

классификатор на основе conversational BERT[1] был обучен на наборе данных Stanford Sentiment Treebank [82] с пятью классами: очень позитивный, позитивный, нейтральный, негативный и очень негативный. Во время использования предсказаний очень позитивные (очень негативные) метки заменялись на позитивные (негативные).

Классификатор токсичности `Toxic Classifier` определяет, содержит ли высказывание оскорбления, угрозы, нецензурные выражения, личную ненависть, откровенные разговоры сексуального характера или другие проявления токсичности. Нейросетевой классификатор на основе conversational BERT-модели[1] был дообучен на наборе данных Kaggle «Toxic Comment Classification Challenge»[2].

Часть диалогового менеджера, называемая `Response Selector`, в боте выбирает окончательный ответ, используя уверенность навыков и оценки от удаленного сервиса `CoBot Conversation Evaluator`. Аннотатор `CoBot Conversation Evaluator` обучен на данных конкурса «Alexa Prize Challenge» от предыдущих конкурсов и предсказывает, будет ли гипотеза интересной, понятной, соответствующей теме, интересной и ошибочной [85]. `CoBot Conversation Evaluator` предоставляется участникам соревнования «Alexa Prize Challenge» как удаленный сервис.

Также используются модели логического следствия от AllenNLP[3], основанные на модели RoBERTa [94], для получения аннотаций MultiNLI [95] и SNLI [96]. Модели логического вывода для пары предложений определяют, являются ли предложения логическим следствием, противоречат или нейтральны по отношению друг к другу.

4.3.2 Корреляция с автоматическими метриками

Итоговые распределения разметки уровней здравого смысла для рассматриваемых навыков на уровне фраз и контекста представлены на Рисунке 4.3 и Рисунке 4.4. Как и ожидалось, основанные на базах знаний

[1] http://docs.deeppavlov.ai/en/master/features/pretrained_vectors.html#downloads
[2] kaggle.com/c/jigsaw-toxic-comment-classification-challenge/overview
[3] demo.allennlp.org/textual-entailment

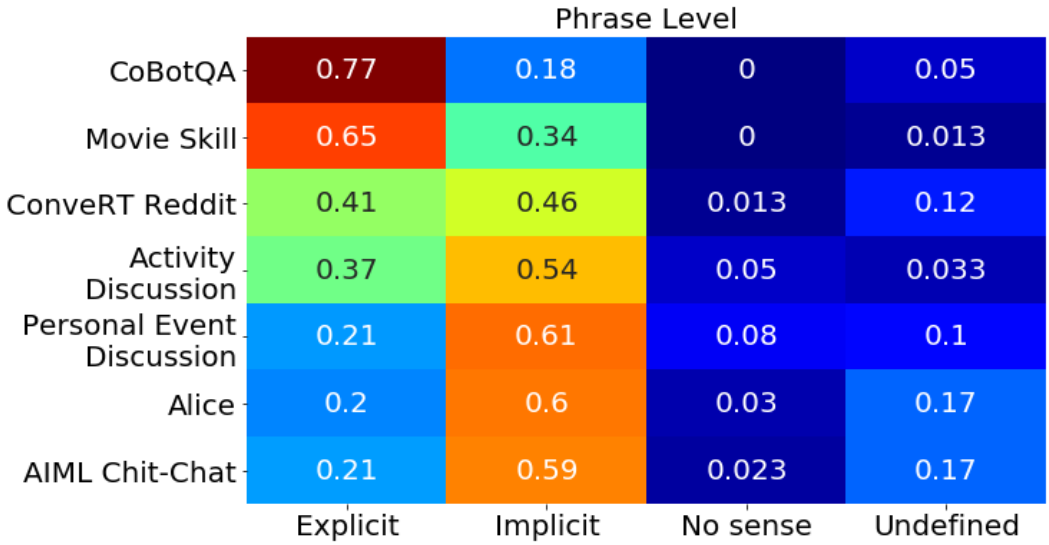

Рисунок 4.3 — Распределение уровня демонстрации здравого смысла на уровне фраз для различных навыков.

`CobotQA` и `Movie Skill` имеют наибольшую долю *explicit commonsense* (*явного здравого смысла*). Для шаблонных навыков общего домена `Alice` и `AIML Chit-Chat` преобладает *implicit commonsense* (*неявный здравый смысл*). Результаты для `Activity Discussion Skill` аналогичны ранжирующему навыку `ConveRT Reddit`. `Personal Event Discussion Skill` – единственный из всех рассмотренных навыков, который имеет более высокую долю *explicit commonsense* (*явного здравого смысла*) на уровне контекста, чем на уровне фразы. Это наблюдение показывает, что по сравнению с другими `Personal Event Discussion Skill` дает наибольшее количество релевантных ответов, которые могут не иметь явного здравого смысла на уровне фраз, но при этом дополняют контекст до явной демонстрации здравого смысла.

На Рисунке 4.5 представлена корреляция размеченных уровней здравого смысла и автоматических метрик. Бессмысленные контексты (размеченные *no sense* и *undefined* на уровне контекста) хорошо характеризуются высоким уровнем токсичности ответов пользователей, низким показателем «понятности» («Comprehensible») по оценке моделью `CoBot Conversation Evaluator` и нейтральным «Neutral» с точки зрения MNLI. Фразы с меткой *no sense* соответствуют низкому уровню «понятности» («Comprehensible»), в то время как *no sense on context level* (*бессмысленные на уровне контекста*) соответствуют нейтральным «Neutral» с точки зрения NLI.

Позитивная реакция пользователя характерна выражению *explicit commonsense* (*явного здравого смысла*) как на уровне фразы, так и на уровне

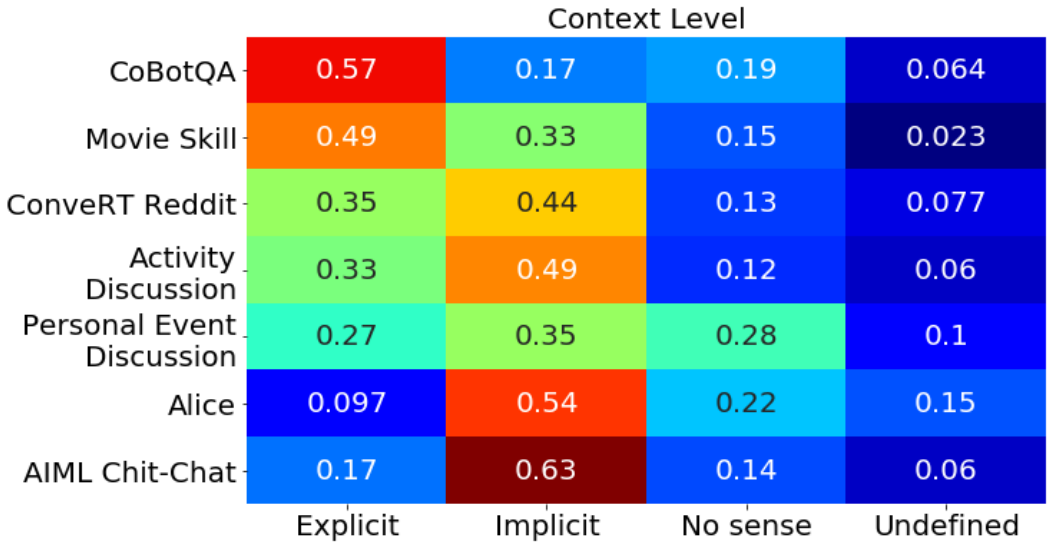

Рисунок 4.4 — Распределение уровня демонстрации здравого смысла на уровне контекста для различных навыков.

контекста, а также имеет значительную отрицательную корреляцию с *no sense* (*бессмысленные*) на уровне контекста.

Параметры `CoBot Conversation Evaluator`, такие как «OnTopic» (соответствует теме), «Interesting» (интересный), и «Erroneous» (ошибочный), могут использоваться для различения *explicit commonsense* и *implicit commonsense* (*явного* и *неявного здравого смысла*) как на уровне фраз, так и на уровне контекста. Параметр логическое следствие «Entailment» в терминах NLI также различает *explicit commonsense* (*явный*) и *implicit commonsense* (*неявный здравый смысл*) на уровне контекста, возможно из-за того, что *явный здравый смысл* подразумевает внесение некоторой новой информации в ответы. Реплики с *implicit commonsense* (*неявным здравым смыслом*) можно охарактеризовать как не содержащие явного здравого смысла, но соответствующие контексту. Это отражается в положительной корреляции с параметрами логического следствия «Entailment» и противоречия «Contradiction».

По результатам, полученным в этой главе, можно сделать следующие **выводы**:

– Была предложена схема разметки уровней наличия здравого смысла на фразовом и контекстном уровнях. В связи с трудозатратностью ручной разметки, было проведено исследование корреляции с автоматическими метриками: тональность, токсичность, оценка реплик в контексте по различным параметрам, модели логического вывода.

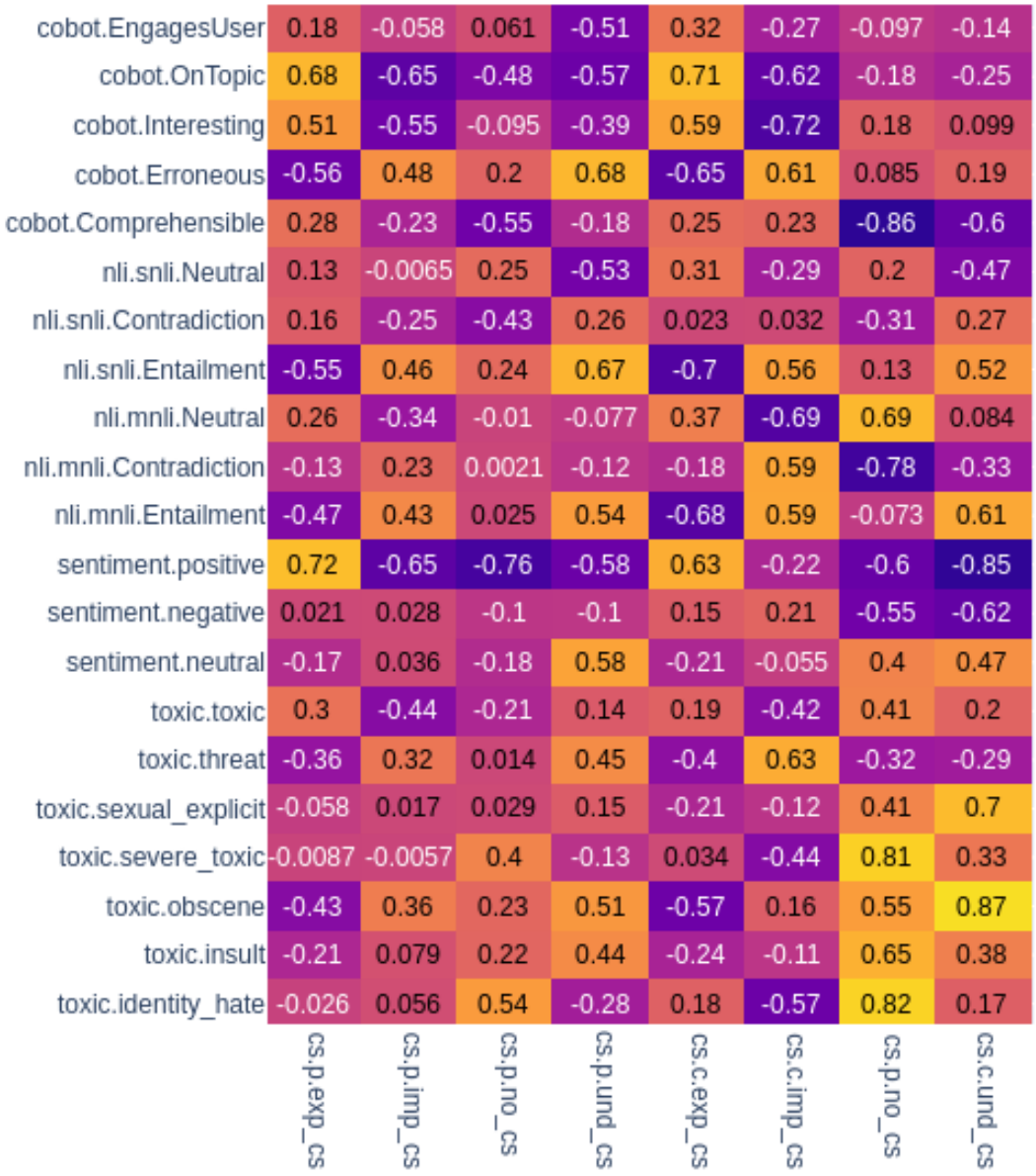

Рисунок 4.5 — Карта корреляции различных видов проявления здравого смысла (*cs.p* – здравый смысл на уровне фраз и *cs.c* – здравый смысл на уровне контекста; явный *exp* и неявный *imp*, неопределенный *und_cs*, отсутствие здравого смысла *no_cs*) и автоматических метрик: тональность «sentiment», токсичность «toxic», логический текстовый вывод «nli» (в частности, «snli» и «mnli»), оценки реплик от `CoBot Conversation Evaluator` «cobot».

- Были представлены два разговорных навыка, интегрирующих модели предсказания здравого смысла в шаблонный подход создания навыков и демонстрирующих более высокий уровень наличия явного здравого смысла, чем шаблонные навыки общего домена, а также имеющих такой же уровень неявного здравого смысла, как и ранжирующие навыки.
- `Personal Event Discussion Skill` имеет самое большое количество ответов, не обладающих явным здравым смыслом на фразовом уровне, но дополняющих контекст до явно выраженного здравого смысла; при этом этот навык имеет самое большое количество бессмысленных реплик. Это означает, что ответы навыка напрямую связаны с контекстом, но при этом могут оказываться бессмысленными из-за неподходящих шаблонов или ошибочных предсказаний здравого смысла.
- «Понятность» («Comprehensible») от `CoBot Conversation Evaluator`, «Нейтральная» («Neutral») метка модели MNLI и уровень токсичности реакции пользователя помогают определить реплики, которые противоречат здравому смыслу или не соответствуют контексту.
- Позитивная тональность реакции пользователя положительно скоррелирована с *explicit commonsense* (*явным здравым смыслом*) на фразовом и контекстном уровнях, при этом имеет значительную отрицательную корреляцию с *no sense* (*бессмысленными*) на уровне контекста.
- Параметры «OnTopic» (соответствует теме), «Interesting» (интересный), and «Erroneous» (ошибочный) `CoBot Conversation Evaluator`, логическое следствие «Entailment» и противоречие «Contradiction» в терминах NLI помогают различить наличие *explicit commonsense* (*явного здравого смысла*) и *implicit commonsense* (*неявного здравого смысла*).
- На предложенные разговорные навыки `Activity Discussion Skill` и `Personal Event Discussion Skill` оформлено свидетельство о государственной регистрации программы для ЭВМ № 2021662601 «Программа разговорных навыков, интегрирующих модели предсказания аспектов здравого смысла в диалоге» [26].
- Программный код предложенных разговорных навыков, интегрирующих модели предсказания аспектов здравого смысла, выложен в открытый доступ[4].

[4] https://github.com/deepmipt/dream

– Исследования, представленные в данной главе, опубликованы в [23].

Исследования, представленные в данной главе, проведены в 2020 году и используют в качестве модели предсказания здравого смысла модель COMeT [91], опубликованную в 2019 году. Другая команда Zotbot, участник «Alexa Prize Challenge 3» также использовала модель COMeT для генерации объекта в триплете для заданных субъекта и отношения. Это привело к тому, что команда Amazon назвала перспективным направление интеграцию моделей предсказания здравого смысла в диалоговые системы открытого домена [12]. Позднее в 2020 году была опубликована новая версия модели COMET-ATOMIC 2020 [97], качество предсказания которой превышает качество использованной в данной работе модели, а также дает возможность получать предсказания для большего числа аспектов. Спустя полгода после публикации оригинальной статьи команды DREAM [21], включавшей методологию использования модели COMeT для генерации реплик в диалоге, была опубликована статья [98] от команды другого университета-участника, в которой было предложено использовать предсказания моделей здравого смысла при генерации реплик, обсуловленной на описание личности бота, для расширения предложений, описывающих персону.

В 2021 году команда CASPR добавила использование предсказаний здравого смысла в бота [99], основанных на комбинации нескольких баз знаний, включая IMDb, Kaggle и другие. В то же время была представлена модель TransOMCS [100], вдвое превосходящая по объему ConceptNet [93] и демонстрирующая возможность переноса лингвистических знаний в суждения здравого смысла. Недавно, в сентябре 2021 года, опубликована работа [101], представляющая набор из 11 тысяч диалогов, в которых демонстрируется здравый смысл. Также в работе представлен подход к автоматической оценке здравого смысла в диалоге, который основан на обучении регрессионной модели на данных ручной оценки здравого смысла в предложенном наборе данных. В отличие от [23], в [101] не выделяются различные уровни демонстрации здравого смысла и нет оценки скоррелированности модели оценки здравого смысла с общедоступными интепретируемыми метриками.

Глава 5. Диалоговый менеджмент

Одной из важнейших частей модульной диалоговой системы является диалоговый менеджер, который фактически управляет диалогом и объединяет в единую систему модели понимания естественного языка и разговорные навыки. Его задачей является не только выбрать подходящую к текущему контексту реплику, но и обеспечить дальнейшее развитие диалога.

В Разделе 5.1 представлена структура диалогового менеджера во фреймворке DeepPavlov Agent. В Разделе 5.2 описан выборщик ответа `Response Selector` диалоговой системы DREAM в рамках конкурса «Alexa Prize Challenge 3», основанный на уверенности навыков. Далее в Разделе 5.3 представлен подход к менеджменту диалога на основе целей. В Разделе 5.4 описан выборщик ответа `Response Selector` диалоговой системы DREAM в рамках конкурса «Alexa Prize Challenge 4», комбинирующий различные виды разговорных навыков, как первый шаг к целеориентированному диалоговому менеджменту.

Подходы и эксперименты, представленные в данной главе, опубликованы в [21; 22; 24].

Автором принимал участие в разработке `Skill Selector` и базовой эвристической модели выбора ответа `Response Selector`. Автором был разработан алгоритм выбора финального ответа на основе тегов, а также проведены эксперименты оценки качества выбора финального ответа, описанные в Разделе 5.4.5. Эксперименты с обучаемыми и ранжирующими моделями выбора финального ответа были проведены членами команды Московского физико-технического института.

5.1 Диалоговый менеджмент DeepPavlov Agent

Во фреймворке DeepPavlov Agent диалоговый менеджер состоит из двух компонент (см. Рисунок 1.5): выборщик навыков `Skill Selector` и выборщик ответа `Response Selector`. `Skill Selector` получает на вход состояние диалога, включая текущую реплику и ее аннотации, и возвращает список

разговорных навыков, которые, согласно алгоритму, могут сгенерировать подходящий ответ. Далее DeepPavlov Agent вызывает независимо друг от друга выбранный набор навыков, каждый из которых может вернуть одну, несколько или ни одной реплик-кандидатов. После этого `Response Selector` получает на вход состояние диалога, включая все аннотированные реплики-кандидаты, и выбирает финальный ответ, который будет возвращен пользователю.

Для выбора навыков **Skill Selector** обычно использует набор правил и условий на историю диалога и аннотации реплики пользователя. `Skill Selector` диалоговой системы DREAM полностью основан на различных условиях на аннотации реплики пользователя. В первую очередь происходит проверка на наличие в реплике пользователя одного из специальных запросов, которые детектируются аннотаторами и требуют определенного действия или реакции от бота, например, когда пользователь спрашивает у бота его имя, по правилам конкурса необходимо соблюдать анонимность и отказываться называть свое имя. Если специальный запрос не обнаружен, то проверяется набор условий для включения «безопасного режима»: не могут быть использованы генеративные навыки, навыки, которые используют слова или фразы из реплики пользователя для заполнения слотов, навыки, выражающие мнение или реплики которых не были проверены на наличие выражение собственного мнения, а также навыки, которые могут давать советы. Это сделано для соблюдения правил соревнования «Alexa Prize Challenge», так как ботам категорически запрещается использовать нецензурную лексику, выдавать неприличные реплики и давать профессиональные советы (медицинские, финансовые и прочие). При этом боту также не рекомендуется выражать мнение по спорным темам, таким как политика или религия, чтобы избежать споров с пользователем, так как это может отразиться на отношении пользователей не только к боту, но и к компании «Amazon» в целом. В конкурсе «Alexa Prize Challenge 3» «безопасный режим» включался в случае наличия нецензурной, токсичной лексики в реплике пользователя или при обнаружении вопроса в реплике пользователя, аннотированной как относящейся к спорной тематике. Однако зачастую пользователи употребляли нецензурную лексику не с целью оскорбить бота, поэтому в «Alexa Prize Challenge 4» «безопасный режим» стал включаться только в случае запроса мнения от пользователя на спорную тему. Это позволило уменьшить случаи прерывания диалога просьбами не употреблять нецензурные выражения.

В Разделе 1.4.1 описаны виды разговорных навыков. Стоит отметить, что в терминологии DeepPavlov Agent каждый навык возвращает в качестве гипотезы не только текст реплики-кандидата, но и численный показатель уверенности (англ.: confidence), а также дополнительные атрибуты, которые могут быть сохранены в атрибуты реплики, бота или пользователя. Ранжирующие и генеративные навыки, которые обычно представлены в виде нейросетевых моделей или моделей машинного обучения, моделью может быть выдан числовой признак для каждой реплики-кандидата при заданном контексте, который может быть использован для получения возвращаемого численного показателя уверенности навыка в реплике-кандидате. Шаблонные навыки обычно не содержат внутри себя нейросетевых моделей или моделей машинного обучения, в связи с чем в них отсутствует возможность автоматически получить показатель уверенности. Поэтому в шаблонных навыках показатель уверенности подбирается вручную разработчиком в зависимости от выполнения различных условий на реплику пользователя и на реплику-кандидата. Например, на определенном шаге сценария шаблонный навык может сначала назначить уровень уверенности по умолчанию, затем понизить или повысить это значение в зависимости от выполнения условий, таких как проверка намерения пользователя, тональности, детектированных сущностей или их типа. Можно снижать уверенность в случаях, когда в реплике пользователя был обнаружен вопрос или когда предыдущая реплика бота подразумевала наличие сущности определенного типа (например, название фильма или животного) в реплике пользователя, но она не обнаружена. Аналогичным образом можно повышать уверенность, если предыдущая реплика бота являлась закрытым вопросом, и текущая реплика пользователя содержит намерение согласия или несогласия.

Рассмотрим в качестве базового варианта **Response Selector**, который выбирает ответ с максимальным значением уверенности навыка в реплике-кандидате. Здесь придется столкнуться с двумя проблемами: (1) выбор ответа в случае нескольких кандидатов с одинаковым значением уверенности, (2) разное происхождение показателей уверенности для различных навыков. Получить некое решение первой проблемы можно путем случайного выбора из лучших реплик-кандидатов или выбора с использованием заданных вручную правил. Вторая проблема же приводит к нечестности выбора ответа на основе уверенности. Действительно, распределения значений уверенности для нейросетевых моделей могут значительно отличаться, а значения уверенности для шаблонных

навыков и вовсе заложены разработчиком. При этом (1) навыки разрабатываются разными авторами, поэтому из-за большого разнообразия сценариев и подходов, вывести единый абсолютный стандарт назначения показателя уверенности не представляется возможным; (2) нельзя гарантировать, что разработчик настраивал показатель уверенности; (3) при выборе финальной реплики среди реплик-кандидатов может быть необходимо учитывать то, какие навыки представили реплики-кандидаты с какими атрибутами (например, если специальный навык для обсуждения определенной темы не выдал гипотезу, то приоритет отдается ранжирующим и генеративным навыкам в расчете на то, что они поддержат тему).

5.2 Выборщик ответа `Response Selector`, основанный на уверенности навыков

В диалоговой системе DREAM в рамках конкурса «Alexa Prize Challenge 3» использовался `Response Selector`, основанный на уверенности навыков. Однако, как было показано в Разделе 5.1, помимо уверенности навыков необходимо также использовать дополнительные модели оценки реплик-кандидатов, которые легко внедряются в качестве аннотаторов реплик-кандидатов. Поэтому использовалась также модель `CoBot Conversation Evaluator` [85], которая, учитывая контекст диалога, выдает для каждого кандидата численные показатели (принимающие значения от 0 до 1) по следующим характеристикам: насколько ответ соответствует теме («isResponseOnTopic»), насколько ответ понятный («isResponseComprehensible»), интересный («isResponseInteresting»), вовлекающий («responseEngagesUser»), ошибочный («isResponseErroneous»). Комбинируя эти параметры, можно сравнивать реплики-кандидаты между собой и выбирать лучшую модель, принимая во внимание такое большое количество характеристик.

В Разделе 3.2 описаны аннотаторы реплик-кандидатов, которые используются в `Response Selector` для выбора финального ответа. В первую очередь для соответствия правилам конкурса «Alexa Prize Challenge», предложенный `Response Selector` отфильтровывает «плохие» реплики-кандидаты с помощью `Toxic Classifier`, `Blacklist Word Detector` и `Dialogue Termination`,

то есть те реплики, которые содержат нецензурные и неприличные выражения, признаки токсичности или имеют высокую вероятностью завершения диалога пользователем после получения данной реплики. Кроме того, если реплика в точности повторяется в контексте диалога, то показатель уверенности у данной реплики-кандидата понижается для повышения лексического разнообразия реплик бота.

Далее `Response Selector` использует базовую эвристическую модель (англ.: Heuristic Baseline) – взвешенную сумму уверенности навыка и предсказаний от аннотатора `CoBot Conversation Evaluator`. Наконец, выбирается ответ с наибольшим значением взвешенной суммы. Формула была получена эмпирическим путем вручную, что не является точным научным подходом, однако в дальнейшем был проведен эксперимент, подтверждающий качество полученной формулы.

Обозначим β – показатель уверенности навыка в реплике-кандидате. Параметры «isResponseOnTopic», «isResponseInteresting», «responseEngagesUser», «isResponseComprehensible», «isResponseErroneous» от `CoBot Conversation Evaluator` обозначены соответственно top, int, eng, com, err. Все шесть параметров принимают вещественные значения от 0 до 1. Также введем параметр γ:

$$\gamma = top + int + eng + com - err. \tag{5.1}$$

Базовая эвристическая формула для вычисления финального показателя α имеет следующий вид:

$$\alpha = 2 * \beta + 0.4 * \gamma. \tag{5.2}$$

Кроме того, предложенный `Response Selector` не ограничивается выбором окончательной реплики только из возможных ответов, но также может составить финальную версию ответа как комбинацию реплик-кандидатов. В «Alexa Prize Challenge 3» эта возможность использовалась для добавления направляющего вопроса для направления диалога в один из сценарных навыков. В самом конце к ответу с некоторой вероятностью может быть присоединено имя пользователя, если оно известно.

Подводя итог, `Response Selector` диалоговой системы DREAM в конкурсе «Alexa Prize Challenge 3» основывался на показателе уверенности и имел следующую верхнеуровневую логику:

1. фильтрация непристойных, нецензурных и токсичных реплик-кандидатов;
2. урезание показателей уверенности за повторы;
3. вычисление финального показателя α для каждой реплики-кандидата с использованием эмпирической формулы;
4. применение заданных вручную эвристик для приоритизации в специальных случаях;
5. выбор финального ответа как реплики с наибольшим показателем α;
6. добавление направляющего вопроса с некоторой вероятностью в случае, если финальная реплика не содержит вопрос и не принадлежит сценарному навыку;
7. присоединение имени пользователя к финальной реплике с некоторой вероятностью, если оно известно.

5.2.1 Эксперименты с моделью выбора финального ответа

Базовая эмпирическая формула была подобрана вручную разработчиками команды DREAM. Такой метод подготовки модели не является научно обоснованным, в связи с чем после накопления достаточного количества разговорных навыков, было принято решение попробовать заменить эмпирическую формулу на обучаемую модель. Для этого были выбраны случайные диалоги с реальными пользователями, в которых реплики-кандидаты были размечены на соответствие контексту для дальнейшего обучения моделей выбора ответа.

Команда DREAM разметила 3400 реплик-кандидатов из $\approx$ 400 уникальных диалоговых контекстов на два класса: подходящий по контексту ответ (положительный) и неподходящий ответ (отрицательный). Диалоги были размечены без перекрытия. Для каждого контекста диалога несколько кандидатов могут быть отмечены как подходящие. В результате был получен набор данных с $\approx$ 750 положительными и $\approx$ 2650 отрицательными примерами. В качестве альтернативы базовой эвристической формуле был применен перебор возможных комбинаций значений параметров (англ.: Grid Search) для настройки коэффициентов перед параметрами β и γ в формуле 5.2 (обозначено как Heuristic Baseline + Grid Search в Таблице 6).

Модель	**Корреляция**
Heuristic Baseline	0.278 ± 0.039
Heuristic Baseline + Grid Search	0.293 ± 0.038
Gradient Boosting	0.326 ± 0.040
Gradient Boosting with TE features	0.335 ± 0.040

Таблица 6 — Результаты экспериментов с моделью выбора ответа в `Response Selector`. Корреляция предсказаний моделей и размеченных вручную меток. Результаты были получены путем усреднения по 500 стратифицированным разбиениям на обучающую и тестовую выборки. TE features обозначает использование признаков из моделей логического вывода.

В связи с наличием размеченных данных для выбора реплик-кандидатов было принято решение продолжить эксперименты по получению обучаемой модели выбора ответа. Для обучения таких моделей выбора ответа были использованы 17 признаков: уверенность навыков в репликах-кандидатах (1), предсказания `CoBot Conversation Evaluator` (5), `Toxic Classifier` (7), `Dialogue Termination` (1) и `Blacklist Words Annotator` (3).

При этом для обучения «лёгких» моделей (в связи с небольшим числом размеченных данных) было необходимо добавить дополнительные числовые признаки. Поэтому был проведен эксперимент с добавлением признаков из моделей логического вывода Textual Entailment (TE), доступных в AllenNLP Demo[1] и определяющих логическое соотношение между предложением-предпосылкой и предложением-гипотезой. В качестве предпосылки использовались две последние реплики диалога, соединенные пробелом, а в качестве гипотезы – реплика-кандидат. Модели логического вывода на выходе предсказывают вероятности принадлежности к трем классам: следствие, противоречие и нейтральное соотношение. В эксперименте было добавлено 9 признаков из 3 моделей логического вывода: Decomposable Attention + ELMo [102] на SNLI [96] (3), RoBERTa [94] на SNLI [96] (3) и RoBERTa [94] на MultiNLI [95] (3).

В качестве модели использовался Gradient Boosting [103] от LightGBM[2]. Результаты, представленные в Таблице 6, показывают, что модели Gradient Boosting превзошли результаты базовой эвристической модели, а предсказания

[1] https://demo.allennlp.org/textual-entailment

[2] https://github.com/microsoft/LightGBM

моделей логического вывода улучшают качество выбора финального ответа. На момент окончания конкурса «Alexa Prize Challenge 3» команда DREAM не использовала модели логического вывода в `Response Selector` из-за значительной вычислительной стоимости использования модели RoBERTa-Large, не сопоставимой с привносимым ею улучшением качества модели выбора ответа.

5.3 Целеориентированный диалоговый менеджмент

Зачастую подходы к отслеживанию состояния диалога и управлению им являются пассивными и одношаговыми, то есть полагаются в основном на предсказываемые диалоговые акты и классификацию намерений последней реплики пользователя. Таким образом, системе не хватает высокоуровневого понимания целей пользователя в диалоге, а значит, не может быть достигнуто полное взаимопонимание между ботом и пользователем. Отчеты команд-участников «Alexa Prize Challenge 3» демонстрируют, что даже поверхностное моделирование понимания пользователя посредством добавления фразы-подтверждения (англ.: acknowledgement) в ответ системы значительно повышает вовлеченность пользователя [16; 17]. Поэтому решение проблемы понимания и разработка стратегий управления диалогом, учитывающих цели пользователя, на момент начала конкурса «Alexa Prize Challenge 4» являлись многообещающим направлением.

Наиболее распространенные реализации отслеживания и управления состоянием диалога работают на уровне одного шага (один шаг диалога – ровно одна итерация обмена репликами между собеседниками). Для того, чтобы сделать управление диалогом более ориентированным на пользователя, предлагается расширить состояние диалога информацией, относящейся к диалоговым целям пользователя. *Диалоговые цели* можно определить как цели пользователя для сегмента диалога. Диалоговые цели могут включать: приветствие, обсуждение темы, смену темы, получение факта, получение совета, обмен опытом и другие. Большинство одно-шаговых целей эквивалентно диалоговым актам, например, приветствие и смена темы, другие темы могут занимать несколько шагов диалога и состоять из нескольких диалоговых актов, например, обсуждение темы или обмен опытом.

Для лучшего установления взаимопонимания и контроля диалога предлагается использовать подход к формированию ответа, мотивированный паттернами, наблюдаемыми в наборе данных Topical Chat [6] и архитектурными особенностями, предложенными другими командами-участниками «Alexa Prize Challenge 3» [12]. Каждый ответ системы будет формироваться как комбинация необязательных частей: фраза-подтверждение (англ.: *acknowledgement*), тело ответа (англ.: *body*) и фраза-затравка (англ.: *prompt*). Фраза-подтверждение (англ.: *acknowledgement*) – это фраза, демонстрирующая понимание того, что сказал пользователь. Тело ответа (англ.: *body*) передает основное содержание всей реплики. Фраза-затравка (англ.: *prompt*) используется для плавного перехода между навыками, темами, сценариями. Фраза-затравка обычно содержит вопрос или предложение, которое предлагает следующую тему и ожидает от пользователя подтверждения переключения диалоговой цели.

Основным навыком в DREAM для проработки взаимопонимания с пользователем стал `Grounding Skill`, предложенный в «Alexa Prize Challenge 4». `Grounding Skill` использует информацию из истории диалога об упомянутых сущностях и определенных намерениях пользователя и бота, чтобы сгенерировать шаблонную фразу-подтверждение для большинства реплик пользователя. Если пользователь переспрашивает, о чем ведется разговор, или иными способами проявляет потерю сути диалога, `Grounding Skill` выдает в качестве основного ответа реплику-подтверждение (англ.: *acknowledgement*). Также *acknowledgement* может быть с некоторой вероятностью добавлен к телу ответа (англ.: *body*), чтобы периодически демонстрировать пользователю понимание предмета обсуждения.

Понимание целей пользователя может улучшить управление диалогом открытого домена. При этом стоит отметить, что зачастую в индустриальных работах по диалоговым системам целеориентированные и задаче-ориентированные диалоговые системы приравниваются друг к другу. В данной работе целеориентированность не отождествляется с задаче-ориентированностью. Пользователь и бот в диалоге на общие темы могут иметь ни одной, одну или несколько целей, которые не являются задачами. Диалоговые цели могут быть осознанными и не осознанными, одношаговыми и многошаговыми. Например, пользователь может заговорить с ботом с целью «поднять себе настроение», которая является неосознанной и при этом, скорее, многошаговой, однако пользователь может не осознавать эту цель и не иметь

задач в диалоге. Поэтому в данной работе сделан фокус на целеориентированность. Целеориентированное управление диалогом может позволить выйти за рамки моделирования диалога как последовательности диалоговых актов и сделать шаг к разработке разговорного искусственного интеллекта, ориентированного на высокоуровневые цели пользователя. Отслеживание диалоговых целей может также улучшить смену темы, переключение между навыками и распознавание намерений. Последовательное знание целей пользователя может позволить диалоговой системе предоставлять своевременные и наиболее релевантные рекомендации для обсуждения. Отправной точкой на пути к разговорному искусственному интеллекту, ориентированному на высокоуровневые цели пользователя, является реализация диалогового менеджера, алгоритм которого представлен в Разделе 5.4. В предлагаемом подходе отсутствует введение понятия целей пользователя и бота, алгоритм основан на сущностях и намерениях пользователя и приоритизации сценарных навыков, как выполняющим многошаговую цель обсуждения темы, сущности.

5.4 Выборщик ответа `Response Selector`, основанный на тегах и комбинирующий различные виды разговорных навыков

Принимая во внимание успех диалоговых систем на основе сценариев [13] и описанный в Разделе 5.3 подход к целеориентированному управлению диалогом, команда DREAM разработала ряд навыков на основе сценариев по популярным темам, чтобы обеспечить контролируемый диалог. Часть навыков (`Movie Skill`, `Book Skill`, `News Skill` и прочие) разработана во время конкурса «Alexa Prize Challenge 3», описана в отчете [21] и Разделе 3.2, обновленные и новые навыки (`Animal Skill`, `Food Skill`, `Sport Skill` и прочие) описаны в отчете [24] и Разделе 3.3. Использование сценариев в диалоге создает впечатление связного диалога «в глубину», но при этом диалоговая система должна уметь «слышать» пользователя и подстраиваться под него в разговоре. Это означает, что диалоговый менеджер должен уметь прерывать сценарий и отдавать приоритет тем навыкам, которые могут удовлетворить текущий запрос пользователя или поддержать резко измененную пользователем тему. То есть с одной стороны, необходимо как можно больше времени проводить пользова-

теля по сценариям и демонстрировать инициативу бота, но с другой стороны, нужно принимать во внимание инициативу пользователя и уметь вовремя переключаться.

Однако пользователь не всегда проявляет инициативу, поэтому диалоговая система также должна иметь встроенную возможность менять тему для дальнейшего развития диалога. Переходы между темами происходят с использованием направляющих вопросов (как описано в Разделах 3.2 и 3.3), которые могут задаваться текущим активным сценарным навыком для осуществления специфичного перехода из навыка в навык, а могут добавляться к выбранной финальной реплике в рамках работы `Response Selector`. Именно он в общем случае принимает решение, стоит ли делать переход на другую тему, а значит, `Response Selector` должен получать информацию не только о текущей уверенности навыка в ответе, но и о возможности выбранного навыка продолжать диалог в дальнейшем.

При использовании вопросов для переключения темы возникает несколько проблем. Излишнее количество направляющих вопросов в ограниченном контексте может создать негативное впечатление о связности бота и его способности к контролю переключения темы. При добавлении направляющего вопроса необходим контроль не только того, что в выбранной финальной реплике не содержится запрос пользователю, но и того, насколько часто пользователю задаются вопросы и насколько давно была смена темы. Однако здесь возникает еще одна проблема, заключающаяся в определении переключения темы в диалоге.

Смена темы в диалоге может происходить как со стороны пользователя, так и со стороны бота. И если известно, что сценарные навыки внутри своего сценария строго поддерживают заданную тему, то другие навыки не могут гарантировать того, что текущая реплика-кандидат не просто подходит по контексту, но и поддерживает и развивает текущую тему. Поэтому `Response Selector` также должен уметь определять насколько реплика-кандидат не сценарного навыка не меняет тему, а наоборот, развивает заданную, так как для большего контроля диалога предпочтительнее менять тему на поддерживаемую сценарными навыками, то есть за счет направляющих вопросов, а не за счет не сценарных реплик, которые могут направить пользователя в уже обсужденную или не покрытую сценариями тему.

Самая важная проблема в описанном в Разделе 5.2 подходе к выбору ответа в рамках конкурса «Alexa Prize Challenge 3» заключается в сильной зависимости от показателя уверенности, который назначается самими навыками разного происхождения. Например, навыки на основе фреймворка AIML имеют одинаковое значение показателя уверенности для всех ответов, в то время как шаблонные и сценарные навыки возвращают показатель уверенности, вручную определенный разработчиком для разных случаев, ранжирующие навыки возвращают показатель сходства в качестве показателя уверенности, а генеративные навыки возвращают выданный нейросетевой моделью показатель уверенности. И не смотря на то, что все показатели уверенности принимают значения от 0 до 1, распределения показателей внутри этого промежутка может сильно отличаться и препятствовать честному сравнению.

Исходя из предположения, что и у пользователя, и у бота есть некоторые высокоуровневые цели в диалоге, как было описано в Разделе 5.3, было разработано и внедрено базовое управление диалогом с учетом целей для диалоговой системы DREAM в рамках конкурса «Alexa Prize Challenge 4», а также заложена основа для более продвинутого управления диалогом с учетом целей в будущих версиях. Поскольку ранжирующие и генеративные модели не могут гарантировать следование целям пользователя и бота, был предложен `Response Selector`, основанный на тегах.

Все реплики-кандидаты распределяются по группам приоритета на основе аннотаций и тегов, а финальная реплика выбирается из группы с наивысшим приоритетом как реплика-кандидат с наибольшим показателем итогового параметра отбора. Первоначально итоговым параметром отбора внутри группы с одинаковым приоритетом было значение показателя α, получаемое с помощью эмпирической формулы, описанной в Разделе 5.2, но позже он был заменен значением, получаемым с помощью обучаемой модели ранжирования реплик-кандидатов, описанной в Разделе 5.4.3.

В Разделе 5.4.1 описаны теги, назначаемые навыками для каждой реплики-кандидата. В Разделе 5.4.2 описан алгоритм приоритизации реплик-кандидатов на основе тегов. Алгоритм финального выбора реплики среди кандидатов одного приоритета представлен в Разделе 5.4.3. И наконец, эксперименты с алгоритмом приоритизации реплик-кандидатов на основе тегов описаны в Разделе 5.4.5.

5.4.1 Теггирование реплик-кандидатов

Модульная диалоговая система DREAM использует комбинацию десятков различных навыков, включая шаблонные, сценарные, ранжирующие и генеративные. Важнейшая особенность диалоговой системы, которая требуется для успешного участия в конкурсе «Alexa Prize Challenge» и заключается в способности поддерживать беседу «в глубину» на популярные темы, в диалоговой системе DREAM отражена в приоритизации сценарных навыков. Поэтому автор предлагает использовать дополнительные теги для каждой реплики-кандидата:

- `Continuation flag` – тег способности навыка продолжить диалог на следующем шаге, если данная реплика-кандидат будет выбрана на этом шаге,
- `Response parts` – тег, содержащий информацию о частях ответа, входящих в реплику.

Оба тега должны быть назначены навыком, который предлагает реплику-кандидата. В случае отсутствия тегов в выходе навыка, присваиваются значения по умолчанию. В итоговом наборе реплик-кандидатов могут быть реплики-кандидаты с одинаковым набором значений тегов.

Continuation flag содержит информацию о способности навыка продолжить разговор на следующем шаге, если текущая реплика-кандидат будет выбрана как финальная и возвращена пользователю. `Continuation flag` может принимать ровно одно из следующих значений:

- `must continue` (должен продолжить) – данная реплика-кандидат с высокой точностью подходит по контексту (по результатам проверки реплики пользователя) и обязательно должна быть возвращена пользователю;
- `can continue prompt` (может продолжить, данная реплика является началом сценария) – данная реплика-кандидат является вступительной репликой сценария для обсуждения определенной темы текущим навыком, и данный навык сможет продолжит диалог, если пользователь поддержит тему;

– `can continue script` (может продолжить) – данная реплика-кандидат является продолжением текущего сценария, но отсутствует высокая точность соответствия контексту, поэтому реплика-кандидат может быть возвращена пользователю в случае, если нет реплик-кандидатов, которые с высокой точностью подходят по контексту (`must continue`);
– `can not continue` (не сможет продолжить) – данная реплика-кандидат не является частью сценария или является финальной репликой в рамках сценария. Является значением по умолчанию.

`Continuation flag` предназначен для приоритизации сценарных навыков, которые способны продолжить диалог, если реплика-кандидат будет выбрана. Не существует гарантированного метода для создания полностью связного диалога, но при этом сценарные навыки могут создать впечатление связности, по крайней мере, на несколько шагов диалога. Направляющие вопросы обозначаются тегом `can continue prompt`, а все навыки, кроме основанных на сценариях, помечаются как `can not continue`. Что касается сценарных и шаблонных навыков, используются следующие правила для выбора значения тега `Continuation flag` (порядок проверок важен):

1. начало сценария:
    а) если пользователь запрашивает диалог на определенную тему
        1) в реплике-кандидате обнаружены шаблоны-триггеры или сущности определенного типа, обозначается `must continue`;
        2) реплика-кандидат принадлежит к запрошенной теме, обозначается `can continue prompt`;
    б) если пользователю был задан направляющий вопрос на данный навык,
        1) в реплике пользователя обнаружены шаблоны-триггеры или сущности определенного типа, или обнаружено ожидавшееся намерение «да»/«нет» пользователя, обозначается `must continue`;
        2) в других случаях, обозначается `can continue prompt`;
    в) в других случаях,

1) если пользователь упомянул шаблоны-триггеры или сущности определенного типа, обозначается `can continue prompt`;
2) в других случаях, если реплика пользователя принадлежит к предлагаемой теме, обозначается `can continue prompt`;

2. внутри сценария:
   а) в реплике пользователя обнаружен ожидавшийся шаблон, обозначается `must continue`;
   б) в реплике пользователя обнаружены шаблоны-триггеры или сущности определенного типа, или обнаружено ожидавшееся намерение «да»/«нет» пользователя, обозначается `must continue`;
   в) во всех других случаях, обозначается `can continue script`;
3. завершение сценария:
   а) в реплике пользователя обнаружены шаблоны-триггеры или сущности определенного типа, или обнаружено ожидавшееся намерение «да»/«нет» пользователя, обозначается `must continue`;
   б) если предыдущая независимая часть сценария закончена, и текущая реплика задает начало новой части сценария с данным навыком (например, когда обсуждение определенного фильма закончено, и навык задает следующий вопрос про кино), обозначается `can continue prompt`;
   в) в других случаях, обозначается `can not continue`.

После внедрения параметра `Continuation flag` было обнаружено, что значение `must continue` в части случаев вводит в заблуждение `Skill Selector`. Параметр `Continuation flag` также используется в `Skill Selector` для обязательного включения сценарного навыка, если его кандидат был выбран в качестве финального ответа на предыдущем шаге и параметр принимает любые значения, кроме `can not continue`. Поэтому если для последней реплики в сценарии проводится достаточная проверка реплики пользователя и назначается тег `must continue` в то время, как сценарий подошел к концу, и навык не будет продолжать диалог (например, осуществляется переход на другой навык), то на следующем шаге `Skill Selector` также включит текущий

сценарный навык в список навыков, которые будут вызваны для генерации реплик-кандидатов. Тем не менее, этот недостаток был оставлен как есть, с расчетом на тот факт, что навык, на который сделан специфичный переход сам вернет реплику-кандидат с тегом `must continue` и перехватит инициативу.

Response parts для каждой реплики-кандидата представляет собой список, обозначающий какие части ответа присутствуют в данной реплике-кандидате. Рассмотрим следующую классификацию составляющих частей ответа:

- `acknowledgement` (подтверждение понимания того, что сказал пользователь) – утверждение, предназначенное для подтверждения понимания ботом того, что сказал пользователь;
- `body` (тело ответа) – основная часть ответа, предназначенная для поддержания текущего разговора;
- `prompt` (затравка) – утверждение или вопрос для начала беседы на новую предлагаемую ботом или запрошенную пользователем тему.

Тег `Response parts` необходим для того, чтобы навык мог передать `Response Selector` информацию, какие составляющие части ответа есть в реплике-кандидате. То есть, содержит ли реплика-кандидат `prompt` для дальнейшего развития диалога, или реплика-кандидат содержит `acknowledgement`. Этот тег используется для объединения нескольких реплик-кандидатов в одну, позволяя, во-первых, расширить реплику путем добавления других частей ответа, а во-вторых, не допустить повторения частей ответа в объединенной реплике. Это связано с тем, что `acknowledgement` может быть использована в реплике навыком, а может быть добавлена в `Response Selector`. То же самое справедливо и для `prompt`.

Направляющие вопросы и вопросы-затравки в сценарных навыках размечаются как `prompt`, остальные реплики-кандидаты по умолчанию отмечаются как `body`, если не указано отдельно автором сценарного навыка. В `Response Selector` вручную задан набор эвристик, которые определяют необходимо ли добавить `prompt` или `acknowledgement` в текущую реплику, отмеченную как `body`.

Специальный шаблонный навык `Grounding Skill` генерирует `acknowledgement` фразы для некоторые видов диалоговых актов пользователя. Например, если пользователь в своей реплике выразил согласие с ботом,

то `Grounding Skill` может составить шаблонный `acknowledgement` в виде «I see you agree. That's cool!» («Вижу, ты согласен. Это классно!»).

5.4.2 Приоритизация реплик-кандидатов на основе тегов

Основной идеей новой версии `Response Selector` стала приоритизация сценарных навыков для обеспечения связного диалога «в глубину» темы. Для этого стало необходимо передавать `Response Selector` больше интерпретируемой информации, чем один вещественный показатель уверенности. Для этого используются аннотации и введены специальные теги, возвращаемые навыками и подробно описанные в Разделе 5.4.1. На основе аннотаций и тегов можно выделить сразу несколько *характеристик, от которых должен зависит приоритет реплики-кандидата*: (1) сценарный ли навык выдал реплику-кандидата, (2) был ли навык активен на предыдущем шаге, (3) пересекаются ли сущности из реплики-кандидата с сущностями из реплики пользователя, (4) какова вероятность, что реплика-кандидат не подходит по контексту, (5) был ли запрос в реплике пользователя, из-за которого стоит прервать сценарий.

Все реплики-кандидаты аннотируются с помощью различных компонент: `CoBot Entities` для извлечения сущностей, `CoBot DialogAct Topics` – тем, `CoBot DialogAct` и `MIDAS Classifier` – диалоговых актов, `Intent Catcher` – намерений, `Dialogue Breakdown` – вероятности несоответствия контексту.

Обозначения Также введем обозначения параметров, значения которых вычисляются с помощью аннотаций и тегов для каждой реплики-кандидата в рамках `Response Selector`, для более короткой записи в дальнейшем:

- `Script` определяет может ли навык продолжить диалог и был ли навык активен на предыдущем шаге:
  - *active* – реплика-кандидат с любым тегом кроме `can continue prompt` от сценарного навыка, который отвечал на предыдущем шаге (то есть реплика продолжает или завершает сценарий),
  - *continued* – реплика-кандидат имеет тег `must continue`, `can continue script` или `can continue prompt`,
  - *finished* – реплика-кандидат имеет тег `can not continue`,

- `Entities` определяет имеет ли реплика-кандидат пересекающиеся сущности с репликой пользователя
  - *same entities* – реплика-кандидат имеет пересекающиеся (по токенам) сущности с репликой пользователя,
  - *other entities* – реплика-кандидат не имеет пересекающихся (по токенам) сущностей с репликой пользователя,
- `Dialogue Breakdown`
  - *no breakdown* – реплика-кандидат соответствует контексту на основе предсказаний аннотатора `Dialogue Breakdown` (определяется по пороговому значению вероятности),
  - *breakdown* – реплика-кандидат не соответствует контексту на основе предсказаний аннотатора `Dialogue Breakdown`.

Базовые приоритеты В Таблице 7 приведено распределение приоритетов в зависимости от значений параметров способности продолжить диалог `Script`, пересечения сущностей `Entities` и вероятности несоответствия контексту `Dialogue Breakdown`. Наивысший приоритет имеют реплики от сценарного навыка, которые продолжает или завершают текущий сценарий. Далее по приоритету расположены реплики-кандидаты, пересекающиеся по упомянутым сущностям с репликой пользователя, и внутри этой группы приоритет имеют реплики от сценарных навыков. Далее приоритет имеют остальные реплики, то есть те, в которых нет пересечения по упомянутым сущностям. В этой группе приоритет также имеют сценарные навыки. Так как вероятность того, что реплика не подходит по контексту вычисляется с использованием нейросетевого аннотатора `Dialogue Breakdown` и может считаться менее точным, чем назначаемые теги, то приоритизация с помощью этого параметра происходит на самом нижнем уровне, то есть когда разбиение по группам приоритета по другим признакам уже произведено.

Частичное использование параметров При этом использование всех предлагаемых признаков также может оказаться избыточным, например, использование модели предсказания вероятности несоответствия контексту может ухудшать выбор финальной реплики. Поэтому рассмотрим случаи выключения использования некоторых признаков. В случае отказа от использования параметра `Entities`, распределение по приоритетам происходит с

Приоритет	Script	Entities	Dialogue Breakdown
I	active	same	no breakdown
II	active	same	breakdown
III	active	other	no breakdown
IV	active	other	breakdown
V	continued	same	no breakdown
VI	continued	same	breakdown
VII	finished	same	no breakdown
VIII	finished	same	breakdown
IX	continued	other	no breakdown
X	continued	other	breakdown
XI	finished	other	no breakdown
XII	finished	other	breakdown

Таблица 7 — Группы приоритетов в зависимости от значения параметров способности продолжить диалог `Script`, пересечения сущностей `Entities` и вероятности несоответствия контексту `Dialogue Breakdown`. Порядок приоритетов указан в столбце «Приоритет», где «I» обозначает высший приоритет.

предположением, что все реплики-кандидаты не имеют пересекающихся сущностей (other entities). В случае отказа от использования параметра `Dialogue Breakdown`, распределение по приоритетам происходит с предположением, что во всех репликах-кандидатах низкая вероятность несоответствия контексту (`no breakdown`).

Инициатива пользователя Представленные в Таблице 7 группы в первую очередь направлены на приоритизацию сценарных навыков. Однако необходимо принимать во внимание тот факт, что пользователи не всегда следуют сценарию, которого придерживается диалоговая система. Зачастую пользователи проявляют инициативу, которая может как проявляться в (1) отказе обсуждать предлагаемую тему, (2) запросе сменить тему, (3) запросе поговорить на определенную тему, (4) резком инициированном пользователем переходе на новую тему, (5) вопросы «в глубину» текущей темы. Первый случай достаточно легко детектируется как отказ или негативная тональность на направляющие вопросы или фразы-затравки из сценариев, например, «Would you like to talk about movies?» – «No. I don't like watching movies.». Прямой запрос на смену

темы, например, «Let's switch the topic.», также с высокой точностью определяется с помощью регулярных выражений или как намерение пользователя. Запрос обсуждения определенной темы хорошо детектируется, а благодаря триггерным словам и выражениям сценарные навыки подхватывают обсуждение, направляя человека в сценарий. Так, например, в качестве запроса на обсуждение музыки будут определены как фраза пользователя «I wanna talk about music.», так и «What are your interests?» – «Music.».

Сложности возникают с обработкой последних двух случаев. Резкий инициированный пользователем переход на новую тему или такие вопросы «в глубину», как «What is your favourite book?» – «I don't know. Do you read often?». При резком переходе (даже не в форме вопроса) на темы, которые также покрыты сценарными навыками (например, «What is your favourite movie?» – «I prefer playing games.»), можно рассчитывать, что соответствующий сценарный навык среагирует с тегом `must continue` благодаря обнаруженным триггерным фразам или сущностям определенного типа. Инициированный пользователем переход на новую тему в не вопросительной форме, когда ни один из других сценарных навыков не обнаружил свои триггеры, может быть легко спутан с ответной репликой по текущей теме. В связи с отсутствием специализированного сценарного навыка и фактическим отсутствием вопроса в реплике пользователя, выгоднее не реагировать на реплику пользователя и оставлять приоритет за текущим сценарным навыком, продолжая диалог.

Однако даже если пользователь не выходит за рамки заданной темы, его реплики могут предполагать необходимость реакции со стороны диалоговой системы. В случаях вопросов «в глубину» заданной темы сценарные навыки не всегда могут ответить на полученный вопрос, а их следующая по сценарию реплика вообще может не подразумевать наличие вопроса в реплике пользователя. Оставить вопрос пользователя без ответа нельзя, так как все впечатление от осознанного сценарного диалога будет испорчено. Поэтому исходя из последних двух случаев проявления инициативы пользователем в вопросительной форме как в рамках текущей темы, так и с переходом на другую тему, необходимо добавить возможность прерывания сценария, то есть лишения текущего сценарного навыка статуса `active`.

Приоритеты в случае вопросов от пользователя Необходимо ввести дополнительные правила приоритизации для случаев, когда пользователь задает

вопросы. Во-первых, сценарный навык, активный на предыдущем шаге, не должен иметь абсолютного приоритета в таких случаях. Во-вторых, ответом на вопрос обычно является утверждение, поэтому предлагается использовать разметку диалоговых актов для определения наибольшего соответствия запросу. Например, когда пользователь запрашивает мнение и реплика аннотируется как «opinion_request» в терминах `MIDAS Classifier`, реплика бота должна выражать мнение и быть аннотирована как «opinion» или «statement» в терминах `MIDAS Classifier`. Таких вопросительных диалоговых актов ограниченное количество, поэтому автор вручную составил карту соответствия диалоговых актов пользователя и бота. В таких случаях добавляется еще один параметр `isRequiredDialogueAct` – аннотирована ли реплика-кандидат ожидаемым пользователем диалоговым актом. Тогда количество групп приоритетов увеличивается вдвое, в первые 12 групп (внутренние приоритеты как в Таблице 7) попадают реплики, которые содержат требуемый диалоговый акт, а вторые 12 групп (внутренние приоритеты как в Таблице 7) приоритета составляют реплики, которые не содержат требуемого диалогового акта.

Если учесть, что `active` значение параметра может быть только у одного навыка, который был выбран в качестве финальной реплики на предыдущем шаге, то становится очевидно, что реплики-кандидаты со значением `active` имеют абсолютный приоритет всегда, кроме случаев, когда используется режим с возможностью прерывания сценария при вопросах пользователя, а также в специальных случаях проявления инициативы.

Алгоритм приоритизации Итак, перейдем к рассмотрению более верхнеуровневого алгоритма `Response Selector`, который использует описанные выше правила приоритизации:

1. если обнаружен **специальный запрос пользователя** (например, пользователь запрашивает недоступную диалоговой системе команду колонки «Alexa»: «Alexa, play music»), наивысший приоритет (тег `active`) отдается специальному навыку `Intent Responder`, который может ответить на такие запросы;
2. если пользователь хочет **сменить тему** или хочет прекратить обсуждать текущую тему, наивысший приоритет (тег `active`) назначается направляющим вопросам, если такие есть среди кандидатов, в противном случае, любым доступным репликам-кандидатам с тегом `prompt`;

3. если пользователь хочет поговорить о какой-то **определенной теме**, наивысший приоритет (тег `active`) назначается тем репликам-кандидатам, которые отмечены тегом `must continue` (сценарные навыки, обнаружившие запрос пользователя на обсуждение относящейся к ним темы, должны выставлять тег `must continue`), следующий приоритет отдается репликам-кандидатам с тегом `prompt`, которые содержат сущности, пересекающиеся с запрошенными пользователем (тема также является извлекаемой сущностью, так как она употреблена в реплике пользователя напрямую). Если таких реплик нет, приоритет отдается любым репликам с тегом `prompt`;
4. если диалоговый акт пользователя **требует определенного действия от бота**, приоритет назначается репликам-кандидатам, содержащим хотя бы один из требуемых диалоговых актов (то есть `isRequiredDialogueAct` $= True$) согласно Таблице 7. В случае отсутствия реплик, содержащих требуемые диалоговые акты, выбирается согласно Таблице 7;
5. в случае отсутствия специальных запросов и для всех остальных реплик-кандидатов, приоритеты распределяются согласно Таблице 7.

Как итог, основной приоритет отдается сценарным навыкам, при этом остается возможность прервать сценарий для удовлетворения специальных запросов или ответа на вопросы пользователя. При наличии нескольких реплик-кандидатов с одинаковым приоритетом, финальный ответ выбирается среди них с помощью ранжирующей модели `Hypotheses Scorer`, описанной в следующем Разделе 5.4.3.

5.4.3 Эксперименты с моделью выбора финальной реплики внутри группы одного приоритета

Финальная реплика выбирается внутри группы с наивысшим приоритетом на основе значений параметра отбора. Параметром отбора изначально было значение α, получаемое с помощью базовой эмпирической формулы. Полученные результаты по обучаемой модели выбора ответа из Раздела 5.2.1 также не были применены в связи с большим количеством изменений, в том числе увеличени-

ем количества сценарных навыков. Поэтому было проведено дополнительное исследование моделей ранжирования реплик-кандидатов.

Наиболее распространены два основных подхода к ранжированию реплик-кандидатов. Первый способ заключается в вычислении независимых векторных представлений контекста и реплики-кандидата, с последующим сравнением представлений для определения релевантности реплики и контекста [48; 104]. Второй подход заключается в определении релевантности на основе совмещенного векторного представления контекста и реплики-кандидата [105—107]. В данной работе было решено попробовать оба варианта, так как каждый имеет свои достоинства и недостатки. Например, использование одной предобученной модели ранжирования возможно без дообучения на собственном наборе данных, а значит экономит время на разметку диалогов. С другой стороны, использование моделей, обученных на реальных диалогах с пользователями, потенциально может давать значительный прирост в качестве.

Эксперименты были проведены в два этапа. На первом этапе для всех пар «контекст-реплика» с помощью предобученной модели ранжирования оценивается релевантность. Полученное значение соответствия контексту добавляется к остальным признакам, полученным от аннотаторов. На втором этапе полученные векторные представления для пар «контекст-реплика» используются для получения итоговых значений релевантности с помощью моделей машинного обучения, обученных на парах «контекст-реплика», размеченных вручную на соответствие контексту.

Для оценки качества моделей ранжирования и для обучения моделей второго этапа были размечены 30 диалогов с реальными пользователями длиной 60-80 реплик. Для каждой реплики пользователя в среднем имеется 6-12 реплик-кандидатов, которые и были размечены на соответствие контексту. В итоге, всего было получено около 10 тысяч примеров, состоящих из контекста, реплики-кандидата и метки соответствия контексту.

Ранжирующие модели могут быть использованы как есть, а могут быть дообучены на диалоговых данных. В связи с применением моделей для диалогового домена, было решено также попробовать дообучить ранжирующие модели. Для дообучения ранжирующих моделей использовался набор диалогов TopicalChat [6], содержащий порядка 10 тысяч диалогов.

Таблица 8 демонстрирует результаты оценки релевантности ранжирующими моделями (первого этапа) на размеченном вручную наборе реальных

диалогов. Нейросетевая модель ConveRT [48], предобученная на наборе данных комментариев с сайта Reddit, генерирует независимые векторные представления для каждой реплики, после чего значение релевантности считается с помощью специальной функции. Другая нейросетевая модель UMS-ResSel [107] основана на нейросетевой архитектуре BERT [5], но использует дополнительные стратегии для процесса обучения. Лучшие результаты ранжирования демонстрирует модель ConveRT, дообученная на наборе диалогов TopicalChat.

Модель	**P@1**	**R@1**	**R@3**	**R@5**	**R@10**
Макс. уверенность	51.2	**50.2**	71.6	85.3	99.0
ConveRT	47.9	46.8	**74.0**	**89.0**	99.1
ConveRT (дообуч.)	**52.6**	49.9	71.6	88.9	**99.6**
UMS-ResSel	47.6	47.6	69.7	84.5	99.0
UMS-ResSel (дообуч.)	48.2	48.0	69.5	85.3	99.1

Таблица 8 — Результаты использования ранжирующих моделей для выбора финального ответа на вручную размеченных реальных диалогах пользователей диалоговой системы DREAM. «Макс. уверенность» выбирает реплику-кандидата с наибольшим значение показателя уверенности навыка. ConveRT – ранжирующая модель на базе архитектуры Transformer, предобученная на наборе комментариев с сайта Reddit. UMS-ResSel – ранжирующая модель на базе архитектуры BERT. «дообуч.» - ранжирующие модели были дообучены на наборе диалогов TopicalChat. В Таблице представлены значения метрик precision P@1 и recall R@K (для топ-K предсказаний).

Во втором этапе экспериментов модели машинного обучения для решения задачи определения соответствия контексту могут использовать все доступные признаки: аннотации, значения релевантности от ранжирующих моделей первого этапа, показатели уверенности навыков. Были использованы предсказания аннотаторов `Dialogue Breakdown` (DB) [108], `MIDAS Classifier` [86], а также предсказания для пяти различных параметров от модели `CoBot Conversation Evaluator`. На данном этапе экспериментов были рассмотрены две имплементации градиентного бустинга: XGBoost [109] и CatBoost [110]. Гипер-параметры были подобраны с помощью поиска по сетке. В Таблице 9 представлены результаты обучения моделей градиентного бустинга с использованием различных

наборов признаков. Базовый набор признаков, использовавшийся во всех экспериментах включает `MIDAS Classifier`, `Dialogue Breakdown`, релевантность от модели ConveRT (дообуч.) из экспериментов первого этапа. В столбце «Доп. признаки» указаны используемые дополнительные признаки. Таблица 9 демонстрирует, что модель XGBRanker, использующая все дополнительные признаки, достигает наилучшего результата.

Модель	**Доп. признаки**	**P@1**	**R@1**	**R@3**	**R@5**	**R@10**
CatBoost	–	63.5	57.0	80.9	93.3	99.7
XGBRanker	Conf	65.7	61.3	83.1	94.3	**99.8**
XGBRanker	Conf + Cobot ConvEval	**68.6**	**63.1**	**84.3**	**94.4**	**99.8**

Таблица 9 — Результаты обучения моделей градиентного бустинга на задаче определения соответствия контексту на размеченных вручную диалогах реальных пользователей. Метрики представлены на валидационной выборке. Модели используют предсказания `MIDAS Classifier`, `Dialogue Breakdown`, релевантность от модели ConveRT (дообуч.), а также дополнительные признаки, указанные в столбце «Доп. признаки». Признак «Conf» обозначает показатель уверенности навыка, а «CoBot ConvEval» – аннотации по пяти параметрам от `CoBot Conversation Evaluator`.

Из проведенных экспериментов можно сделать следующие выводы: (1) использование ранжирующих моделей без дообучения не превосходит по результатам использование модели, выбирающей финальный ответ максимизацией уверенности навыка; (2) дообучение ранжирующих моделей на данных соответствующего домена (в данном случае, диалогового) улучшает качество ранжирования; (3) использование моделей машинного обучения, которые обучены на большом количестве дополнительных признаков на вручную размеченном наборе реальных диалогов, значительно повышает качество ранжирования реплик-кандидатов.

5.4.4 Комбинация реплик-кандидатов

`Response Selector` не только выбирает финальную реплику как одну из реплик-кандидатов, но и может объединять их между собой. Это связано с необходимостью добавления `acknowledgement` и `prompt` частей к финальной реплике для демонстрации понимания пользователя и направления дальнейшего развития диалога.

Для этого, во-первых, в `Response Selector` есть возможность добавить `acknowledgement` к финальной реплике, если ее соответствующий тег указывает, что она не содержит `acknowledgement` часть. Добавление происходит с определенной вероятностью, чтобы не употреблять `acknowledgement` слишком часто. В основном, реплики-кандидаты с тегом `acknowledgement` генерируются с помощью шаблонов и специальных правил на основе текущего диалогового акта пользователя и упомянутых сущностей в `Grounding Skill`, задачей которого является установление взаимопонимания с пользователем. В дальнейшем планируется добавить `acknowledgement` на основе определенных аннотаторами настроения и эмоций пользователя, а также предсказанной эмоциональной реакции диалоговой системы.

Во-вторых, как было сказано ранее, одной из важных функций `Response Selector`, как составляющей диалогового менеджмента бота, является развитие диалога. Поэтому направляющие вопросы, которые генерируются в рамках отдельного навыка на основе рекомендаций следующей темы в диалоге, могут быть добавлены к финальной реплике, если таковая не содержит в себе вопросов, тега `prompt`, а также не является промежуточной репликой сценария. Реплика добавляется с некоторой вероятностью, при этом также учитывается то, когда в последний раз был задан направляющий вопрос, если на данный момент активным является не сценарный навык, так как одной из целей `Response Selector` является приоритизация сценарных навыков для улучшения контроля над диалогом.

5.4.5 Эксперименты с условиями приоритизации реплик-кандидатов на основе тегов

Алгоритм выбора финального ответа имеет несколько важных условий приоритизации реплик-кандидатов, описанных в Разделе 5.4.2. Есть несколько условий, которые могут быть как включены для использования в алгоритме приоритизации, так и отключены: (1) предсказания модели определения вероятности несоответствия реплики-кандидата контексту `Dialogue Breakdown`; (2) приоритизация реплик-кандидатов, имеющих общие сущности с репликой пользователя; (3) прерывание сценариев в случае запросов от пользователя. Использование модели определения вероятности несоответствия реплики-кандидата контексту может позволить дать приоритет репликам-кандидатам, которые соответствуют контексту. приоритизация реплик-кандидатов, имеющих общие сущности с репликой пользователя, фактически повышает приоритет репликам-кандидатам, которые за счет наличия общих сущностей потенциально являются релевантными контексту, так как упоминают объект, названный пользователем. Использование сценарных навыков может помочь улучшить связность диалога, однако сценарии обычно не предусматривают случаи, когда пользователь проявляет инициативу и задает вопросы, поэтому возможность прерывания сценариев в случае запросов от пользователя может позволить лучше реагировать на инициативу пользователя. Поэтому в данном разделе проводится исследование, как 3 заявленных условия приоритизации влияют на выбор финального ответа.

Для этого был проведет эксперимент, состоящий из 4 этапов:

1. сбор диалогов с реальными пользователями с помощью краудсорсинговой платформы Yandex Toloka[3];
2. разметка для каждого набора *контекст диалога + реплики-кандидаты* финальных реплик, выбираемых с помощью алгоритмов приоритизации с разными условиями;
3. для всех случаев, когда ответы, выбранные различными алгоритмами, отличаются, разметка с помощью краудсорсинговой платформы Yandex Toloka[3], определяющая для каждой пары *контекст диалога*

[3] https://toloka.yandex.ru

+ финальная реплика-кандидат соответствует ли финальная реплика контексту;

4. сравнительный анализ для собранных данных для разных версий алгоритма приоритизации.

Диалоги с пользователями «Alexa Prize Challenge» являются приватными и не могут быть использованы для разметки с помощью краудсорсинговых платформ. Поэтому для получения диалогов для дальнейшей разметки был произведен сбор диалогов пользователей с диалоговой системой DREAM. В качестве диалогового интерфейса использовался мессенджер Telegram[4]. На первом этапе было собрано более 378 диалогов с пользователями Toloka, из них были отобраны 4793 набора *контекст + реплики-кандидаты*, которые были размечены различными версиями алгоритма выбора финальной реплики. Все наборы *контекст + реплики-кандидаты* были размечены 9 версиями `Response Selector`, в алгоритме 8 версий были использованы все различные комбинации включения 3 заявленных условий, и дополнительно базовым эвристическим алгоритмом, представленным в Разделе 5.2. Для определенного контекста для некоторых комбинаций условий выбираемые алгоритмами финальные реплики могут совпадать, поэтому для экономии ресурсов размечались только уникальные пары *контекст + финальная реплика-кандидат* и только для контекстов, для которых различными алгоритмами было выбрано не менее двух различных финальных реплик. Для 4339 контекстов все алгоритмы выдали одинаковые финальные реплики. На Рисунке 5.1 среди отобранных 4793 наборов *контекст + реплики-кандидаты* в почти 70% примеров все версии `Response Selector` выдали одинаковый ответ. В итоге, для разметки на соответствие контексту было отобрано 2457 наборов *контекст + финальная реплика-кандидат*.

Для третьего этапа эксперимента, была проведена разметка на соответствие финальной реплики контексту с помощью краудсорсинговой платформы Yandex Toloka. Перед англо-язычными разметчиками была поставлена задача определить, подходит ли реплика к заданному контексту: был задан вопрос «Does the response suit the context?» и два варианта ответа «Response suits the context» и «Response does NOT suit the context». На Рисунке 5.2 показан пример 1 задания. На каждой странице, показываемой пользователю, было представлено $N = 5$ заданий, что означает, что каждый пользователь размечал не менее 5 примеров при $L = 2$ возможных классах. Также все задания

[4] https://telegram.org/

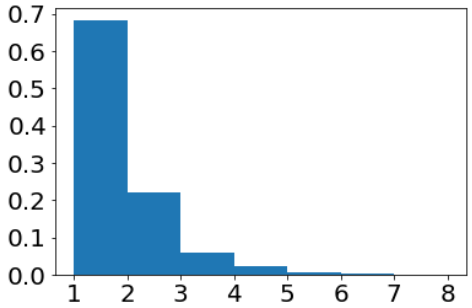

Рисунок 5.1 — Распределение 4793 наборов *контекст + реплики-кандидаты* по количеству различных уникальных финальных ответов, выбранных 8 различными версиями `Response Selector` и базовым эвристическим алгоритмом, представленным в Разделе 5.2.

размечались с показателем перекрытия 5, то есть каждое задание размечалось 5 разными асессорами. При разметке использовались дополнительные фильтры, чтобы обеспечить разнообразие разметчиков (разметка не более 3 страниц, то есть 15 заданий от одного асессора) и достаточное качество разметки (проект был доступен только 20% лучших асессоров по метрикам Yandex Toloka). Также были добавлены контрольные задания, которые представляют из себя 40 наборов *контекст + финальная реплика-кандидат*, размеченных автором диссертации, при ошибках в разметке которых пользователям блокировалось участие в проекте.

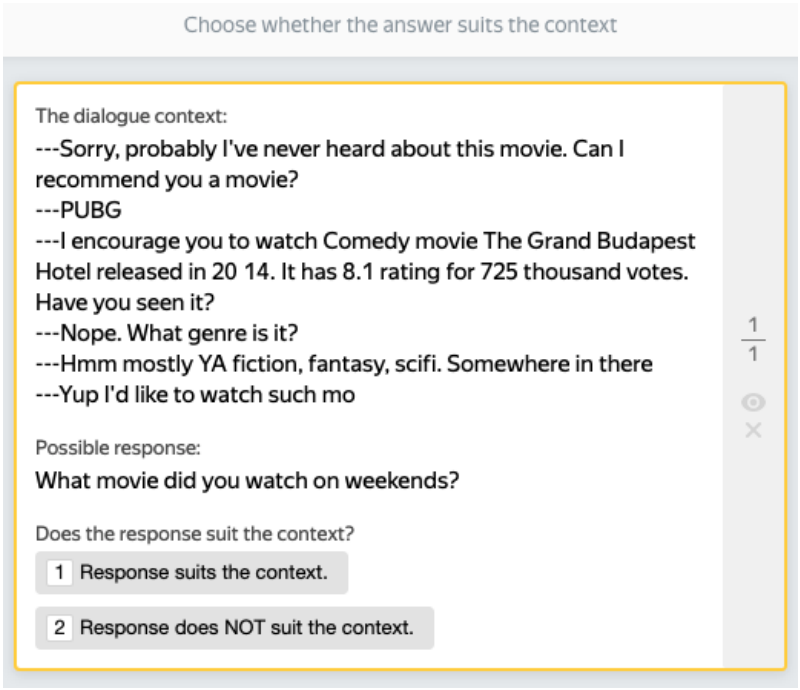

Рисунок 5.2 — Пример задания с краудсорсинговой платформы Yandex Toloka для разметки на соответствие финальной реплике контексту.

Так как минимальное число размечаемых заданий каждым пользователем больше квадрата количества классов разметки $N > L^2$, может быть применен метод агрегации[5] Дэвида-Скина [111], который для каждого разметчика оценивает L^2 параметров. Программная реализация метода агрегации Дэвида-Скина находится в открытом доступе[6] и автоматически может быть применена к результатам разметки на платформе Yandex Toloka. Собранные данные разметки с перекрытием 5 были обработаны алгоритмом агрегации Дэвида-Скина, в результате чего были получены 2457 примеров, где для каждого примера были приведены итоговый размеченный класс и показатель значимости ответа в процентах.

В Таблице 10 приведены результаты агрегации данных, размеченных на соответствие финальной реплики, выбранной разными алгоритмами приоритизации `Response Selector`. Доли реплик, подходящих по контексту, подсчитаны среди всех случаев, когда разные версии алгоритма выбрали хотя бы две различных финальных реплики. В столбцах приведены индикаторы включения «+» и отключения «—» приоритизации по обозначенным условиям: (1) предсказания модели определения вероятности несоответствия реплики-кандидата контексту `Dialogue Breakdown`; (2) общие сущности реплики-кандидата с репликой пользователя; (3) прерывание сценария в случае запросов от пользователя. Приведены результаты для ответов с уровнем значимости $> 50\%$ и $> 95\%$, то есть при подсчете доли реплик, соответствующих контексту, учитывались только агрегированные ответы с уровнем значимости больше заданного порога. Результаты приведены в возрастающем порядке. При попарном сравнении каждой комбинации 3 рассматриваемых условий, которые отличаются включением-отключением одного из условий, видно, что включение условия увеличивает долю реплик, размеченных как соответствующие контексту. Включение всех 3 рассматриваемых условий приоритизации дает самый лучший результат — доля реплик, соответствующих контексту с уровнем значимости $> 95\%$, составляет 0.64, что значительно превосходит 0.42 долю реплик, подходящих по контексту, при отключении всех 3 рассматриваемых условий приоритизации. Это значит, что все 3 предложенных условия приоритизации в алгоритме `Response Selector` являются полезными и улучшают качество выбора реплики по критерию соответствия контексту. Причем, предложенный алгоритм выбо-

[5] https://yandex.ru/support/toloka-requester/concepts/result-aggregation.html
[6] https://github.com/Toloka/crowd-kit/blob/main/src/aggregation/dawid_skene.py

ра финального ответа `Response Selector` при использовании всех заявленных условий превосходит по качеству базовый эвристический алгоритм, представленный в Разделе 5.2.

#	Модель прерывания	Общие сущности	Прерывание сценария	Значимость > 50%		Значимость > 95%	
				Число примеров	Доля реплик, подходящих по контексту	Число примеров	Доля реплик, подходящих по контексту
0	Базовая модель			1704	0.472	1402	0.486
1	—	—	—	641	0.417	517	0.424
2	+	—	—	640	0.442	513	0.450
3	—	—	+	911	0.490	736	0.496
4	—	+	—	637	0.516	516	0.537
5	+	+	—	635	0.532	514	0.551
6	+	—	+	632	0.560	507	0.574
7	—	+	+	637	0.609	515	0.631
8	+	+	+	632	**0.620**	513	**0.641**

Таблица 10 — Результаты агрегации разметки финальных реплик на соответствие контексту. Показатели значимости ответа возвращаются алгоритмом агрегации Дэвида-Скина для каждого примера. В столбцах приведены количество примеров и доли реплик, подходящих по контексту, для разных пороговых значений показателя значимости. Финальные реплики получены с помощью алгоритма приоритизации с разными комбинациями используемых условий. Доли реплик приведены среди всех случаев, когда разные версии алгоритма выбрали хотя бы две различных финальных реплики. Модель прерывания – использование классификатора `Dialogue Breakdown` для получения вероятности соответствия реплики-кандидата контексту. Общие сущности – реплики-кандидаты, имеющие общие сущности с последней репликой пользователя. Прерывание сценария – прерывание сценария в случае запроса от пользователя. Базовая модель – базовый эвристический алгоритм выбора ответа, представленный в Разделе 5.2.

5.5 Другие подходы к диалоговому менеджменту

Архитектура диалогового менеджмента в системе DREAM обусловлена использованием библиотеки `DeepPavlov Agent`, который подразумевает две главные составляющие диалогового менеджера: выборщик навыков `Skill Selector` и выборщик ответов `Response Selector`. В системе DREAM используется `Skill Selector`, основанный на проверке различных условий на аннотации реплики пользователя и историю диалога. Простота алгоритма объясняется возможностью выбрать набор навыков «с запасом» для текущего контекста. Куда более сложная функция возложена на `Response Selector`, который выбирает (в некоторых случаях, комбинирует) финальную реплику среди реплик-кандидатов. Задача выбора реплики для заданного контекста среди заданного набора реплик-кандидатов представляет из себя задачу ранжирования, которая можно решать с помощью предобученных ранжирующих моделей либо с помощью обучаемых моделей машинного обучения. В диссертации представлены эксперименты по использованию и сравнению различных предобученных ранжирующих и обучаемых моделей выбора ответа.

Тем не менее участие диалоговой системы DREAM в международном конкурсе диалоговых систем общего домена с постоянным отслеживанием пользовательского рейтинга накладывало определенные ограничения. Задача поддержания рейтинга на как можно более высоком уровне привела к необходимости контроля не только за качеством отдельных реплик, но и за развитием и качеством каждого диалога в целом, включая выбор следующей темы и плавный переход к ней. Как было показано в предыдущем конкурсе [13], сценарные навыки являются важным элементов диалога, который позволяет контролируемо развивать диалог «в глубину» темы. В связи с этим в алгоритм выбора финального ответа было введено использование различных правил приоритезации, а внутри группы одного приоритета реплика выбирается согласно показавшей наилучшее качество обучаемое модели выбора финального ответа, использующей в качестве признаков показатели уверенности навыков, векторные представления из лучшей модели ранжирования, распределение вероятностей по диалоговым актам, а также предсказания модели оценки реплики в контексте.

Другие диалоговые системы-участники конкурса использовали различные варианты диалогового менеджмента в соответствии с общей архитектурой системы. Например, системы, основанные на диалоговых графах, в качестве диалогового менеджера использовали конечные автоматы [64] и treelets [17]. Несколько команд использовали похожий на DREAM подход, заключавшийся в выборе набора навыков (также называемых генераторами) и выборе финального ответа. Алгоритмы выбора в большинстве своем основаны на проверке набора различных условий.

Таким образом, алгоритм диалогового менеджера зависит от общей архитектуры системы. У большинства команд-участников алгоритм диалогового менеджера основан на наборе различных правил и условий, несмотря на то, что были попытки сделать его обучаемым [64; 112].

По результатам, полученным в данной главе, можно сделать следующие **выводы**:

- Реализованы алгоритмы выбора финального ответа `Response Selector` для диалоговой системы DREAM, основанный на уверенности навыков и основанный на тегах.
- Для алгоритма выбора финального ответа, основанного на тегах, проведен сравнительный анализ использования условий приоритизации реплик с помощью краудсорсинговой платформы Yandex Toloka.
- Предложенный алгоритм `Response Selector` на основе тегов позволил улучшить выбор финального ответа (доля соответствующих контексту реплик возросла на 15.5% по сравнению с базовым эвристическим алгоритмом).
- Использование трех предложенных условий приоритизации реплик-кандидатов позволяет улучшить результаты выбора финального ответа (доля соответствующих контексту реплик увеличилась более, чем на 20% по сравнению с версией, не использующей предложенные условия). Доли реплик считаются среди всех случаев, когда разные версии алгоритма выбрали хотя бы две различных финальных реплики.
- Результирующая модель, полученная в экспериментах с обучаемыми моделями выбора финального ответа, была интегрирована в качестве модели выбора финальной реплики внутри группы одного приоритета в `Response Selector`.

– На предложенный алгоритм выбора финального ответа оформлено свидетельство о государственной регистрации программы для ЭВМ № 2021662460 «Программа выбора финального ответа из реплик-кандидатов» [27].
– Программный код обеих версий алгоритма выбора финального ответа, предложенных в данной главе, выложен в открытый доступ[7].
– Технические описания двух версий алгоритма выбора финального ответа, представленных в данной главе, опубликованы в [21; 22; 24].

[7] https://github.com/deepmipt/dream

Заключение

Основные результаты работы заключаются в следующем.

1. Исследование зависимости качества решения задачи классификации текстов от доменной специфичности, в частности языкового стиля, показало, что использование векторных представлений соответствующего целевой задаче домена позволяет улучшить результаты для задач классификации текстов (до 3.6 пунктов F-1 метрики на 5 разных задачах). Предложенный подход был применен к построению классификаторов для диалоговой системы DREAM.
2. Предложенный метод интеграции нейросетевых моделей предсказания здравого смысла повышает количество высказываний бота, содержащих или дополняющих контекст диалога до здравого смысла. Предложенные разговорные навыки, использующие данный метод, были интегрированы в диалоговую систему DREAM испытаны и применены на реальных пользователях.
3. В соответствие с предложенной схемой разметки уровней здравого смысла в диалогах, был размечен закрытый набор реальных диалоговых данных, проведено исследование корреляции предложенной разметки здравого смысла и нескольких автоматических метрик.
4. Исследование корреляции представленной разметки здравого смысла и автоматических метрик показало, что тональность и токсичность реакции пользователя скоррелированы с явными проявления здравого смысла и полным отсутствием здравого смысла.
5. Предложенные алгоритмы управления диалогом интегрированы в компоненты `Skill Selector` и `Response Selector` диалоговой системы DREAM в конкурсах «Alexa Prize Challenge 3» и «Alexa Prize Challenge 4».
6. Предложенный алгоритм `Response Selector` на основе тегов позволил улучшить выбор финального ответа. Доля реплик, соответствующих контексту, возросла на 15.5% по сравнению с базовым эвристическим алгоритмом и более, чем на 20% по сравнению с версией, не использующей предложенные условия.

7. Исходный код всех разработанных в рамках данной работы моделей и программ опубликован в открытом доступе, как часть библиотеки `DeepPavlov`[8] и диалоговой системы `DREAM`[9].

[8] http://docs.deeppavlov.ai/en/master/features/models/classifiers.html

[9] https://github.com/deepmipt/dream

Список сокращений и условных обозначений

AIML	язык разметки искусственного интеллекта [Artificial Intelligence Markup Language]
ASR	модуль распознавания речи [Automatic Speech Recognition]
BERT	векторные представления из двунаправленных кодировщиков, имеющих архитектуру Трафнсормер [Bidirectional Encoder Representations from Transformers]
BiGRU	двунаправленная нейросетевая модель, содержащая управляемый рекуррентный блок [Bidirectional Gated Recurrent Unit]
BiLSTM	двунаправленная нейросетевая модель, обладающая долгой краткосрочной памятью [Bidirectional Long Short-Term Memory]
BiLSTM-CRF	двунправленная нейросетевая модель, обладающая долгой краткосрочной памятью с условным случайным полем [Bidirectional Long Short-Term Memory with a Conditional Random Field]
CNN	свёрточная нейронная сеть [Convolutional Neural Network]
CoBot	закрытый фреймворк для построение диалоговых систем от Amazon [Conversational Bot]
ConveRT	векторные представления разговорного стиля, получаемые из моделей архитектуры Трансформер [Conversational Representations from Transformers]
CoVe	векторные представления контекстов целиком [Context Vectors]
DFF	среда для создания сценарных навыков [Dialog Flow Framework]
DM	диалоговый менеджер [Dialogue Management]
ELMo	векторные представления из языковых моделей специфичной архитектуры [Embeddings from Language Models]
GloVe	векторные представления слов, обучаемые на основе совстречаемости слов в корпусе [Global Vectors]
InferSent	метод получения векторных представлений предложений целиком для английского языка

LSTM	нейросетевая модель, обладающая долгой краткосрочной памятью [Long Short-Term Memory]
MNLI	набор данных разных доменов, где каждый пример представляет из себя пару предложений и метку, являются ли предложения логическим следствием, противоречат или нейтральны по отношению друг к другу [The Multi-Genre Natural Language Inference Corpus]
NER	распознавание именованных сущностей [Named Entity Recognition]
NLG	модуль генерации естественного языка [Natural Language Generation]
NLU	модуль понимания естественного языка [Natural Language Understanding]
SNLI	набор данных, где каждый пример представляет из себя пару предложений и метку, являются ли предложения логическим следствием, противоречат или нейтральны по отношению друг к другу [The Stanford Natural Language Inference Corpus]
SST	Стенфордский датасет определения тональности [Stanford Sentiment Treebank]
STDM	диалоговый менеджер, основанный на переходах из состояния в состояние [State Transition Dialogue Manager]
SWCNN	неглубокая широкая свёрточная нейронная сеть [Shallow-and-Wide Convolutional Neural Network]
TagLM	модели классификации элементов последовательности на векторных представлениях языковой модели [Language-Model Augmented Sequence Tagger]
TF-IDF	векторное представление важности слова в контексте документа, который является частью коллекции документов [Term Frequency, Inverse Document Frequency]
TTS	модуль генерации речи [Text-to-speech]
UMS-ResSel	Стратегии управления репликами для выбора ответа [Utterance Manipulation Strategies for Response Selection]
XML	расширяемый язык разметки [eXtensible Markup Language]

Словарь терминов

Векторные представления слов/текстов (Embedding) : представление слова/текста в виде вектора фиксированной длины с вещественными значениями.

N-грамма : последовательность из N элементов (символов, слов).

Токен (Token) : текстовая единица, представляющая из себя слово целиком или N-грамму символов.

Языковая модель (Language Model) : нейросетевая модель, обученная для решения задачи моделирования языка, то есть предсказания следующего слова/токена в тексте.

Тональность (Sentiment) : эмоциональная окраска текста. Обычно выделяют позитивную, негативную, нейтральную.

Токсичность (Toxicity) : вид негативной характеристики текста, обычно означает наличие в тексте нецензурных выражений, оскорблений, непристойностей, личностной ненависти и пр.

Разговорный навык (Conversational Skill) : модель, производящая по заданном контексту реплику-гипотезу, которая может являться продолжением диалога.

Здравый смысл (Commonsense) : совокупность взглядов и знаний, используемых человеком в повседневной жизни, которые принимаются окружающими людьми по умолчанию.

Домен (Domen) : специфичность, область применимости или происхождения.

Ранжирующий навык (Retrieval Skill): модель, извлекающая реплику-гипотезу по заданному контексту из заданного набора возможных реплик методом ранжирования.

Поиск по сетке (Grid Search) : алгоритм подбора гипер-параметров, основанный на оценке качества модели для каждой комбинации гипер-параметров и выборе лучшей комбинации.

Фрейм (Frame) : это некая структура знаний, представляющая информацию, которую система может извлечь из реплик пользователя, и состоит из набора слотов, каждый из которых может принимать значения из заданного набора.

Цельная диалоговая система (End-to-end dialogue System) : система, состоящая из одной модели, которая получает на вход текст реплики и генерирует финальный ответ.

Модульная диалоговая система (Module-based Dialogue System) : система, состоящая из нескольких компонент, которая получает на вход текст реплики и генерирует финальный ответ.

Задаче-ориентированная диалоговая система (Task-oriented Dialogue System): система, диалог с который ведет для выполнения некоторой задачи.

Состояние диалога (Dialogue State) : это структурированная информация, содержащая в себе историю диалога, включая реплики-кандидаты, аннотации всех реплик и реплик-кандидатов, а также специальные атрибуты пользователя и системы.

Аннотаторы (Annotators) : набор моделей понимания естественного языка, который получает на вход текст реплики и состояние диалога, обычно включают в себя исправление опечаток, различные виды классификации текста и токенов, извлечение сущностей, а также другие модели анализа текста.

Выборщик навыков (Skill Selector) : компонента, формирующая список навыков, которые будут вызваны для генерации реплик-кандидатов.

Выборщик ответа (Response Selector) : компонента, использующая состояние диалога, реплики-кандидаты и их аннотации для выбора финального ответа, возвращаемого пользователю.

Коэффициент Каппа Коэна (κ) : это статистика, которая используется для измерения надежности между экспертами (а также надежности внутри экспертов) при разметке категориальных признаков.

Фраза-подтверждение (Acknowledgement) : фраза, демонстрирующая понимание того, что сказал пользователь.

Фраза-затравка (Prompt) : фраза, использующаяся для плавного перехода между навыками, темами, сценариями.

Yandex Toloka : краудсорсинговая платформа https://toloka.yandex.ru.

Telegram : мессенджер https://telegram.org/.

Список литературы

1. *Weizenbaum, J.* ELIZA—a computer program for the study of natural language communication between man and machine / J. Weizenbaum // Communications of the ACM. — 1966. — Т. 9, № 1. — С. 36—45.
2. *Dai, A. M.* Semi-supervised sequence learning / A. M. Dai, Q. V. Le // Advances in neural information processing systems. — 2015. — С. 3079—3087.
3. Deep Contextualized Word Representations / M. Peters [и др.] // Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers). — 2018. — С. 2227—2237.
4. Improving language understanding by generative pre-training / A. Radford [и др.] // URL https://s3-us-west-2. amazonaws. com/openai-assets/research-covers/languageunsupervised/language understanding paper. pdf. — 2018.
5. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding / J. Devlin [и др.] // Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). — Minneapolis, Minnesota : Association for Computational Linguistics, 06.2019. — С. 4171—4186. — URL: https://www.aclweb.org/anthology/N19-1423.
6. Topical-Chat: Towards Knowledge-Grounded Open-Domain Conversations. / K. Gopalakrishnan [и др.] // INTERSPEECH. — 2019. — С. 1891—1895.
7. Personalizing Dialogue Agents: I have a dog, do you have pets too? / S. Zhang [и др.] // Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). — 2018. — С. 2204—2213.
8. Wizard of wikipedia: Knowledge-powered conversational agents / E. Dinan [и др.] // arXiv preprint arXiv:1811.01241. — 2018.
9. *Reddy, S.* Coqa: A conversational question answering challenge / S. Reddy, D. Chen, C. D. Manning // Transactions of the Association for Computational Linguistics. — 2019. — Т. 7. — С. 249—266.

10. Quac: Question answering in context / E. Choi [и др.] // arXiv preprint arXiv:1808.07036. — 2018.

11. The design and implementation of xiaoice, an empathetic social chatbot / L. Zhou [и др.] // Computational Linguistics. — 2020. — Т. 46, № 1. — С. 53—93.

12. Further Advances in Open Domain Dialog Systems in the Third Alexa Prize Socialbot Grand Challenge / R. Gabriel [и др.] // Alexa Prize Proceedings. — 2020.

13. Emora: An inquisitive social chatbot who cares for you / S. E. Finch [и др.] // arXiv preprint arXiv:2009.04617. — 2020.

14. *Niven, T.* Probing Neural Network Comprehension of Natural Language Arguments / T. Niven, H.-Y. Kao // Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. — Florence, Italy : Association for Computational Linguistics, 07.2019. — С. 4658—4664. — URL: https://www.aclweb.org/anthology/P19-1459.

15. *Marcus, G.* GPT-2 and the Nature of Intelligence / G. Marcus // The Gradient. — 2020.

16. Gunrock 2.0: A user adaptive social conversational system / K. Liang [и др.] // arXiv preprint arXiv:2011.08906. — 2020.

17. Neural generation meets real people: Towards emotionally engaging mixed-initiative conversations / A. Paranjape [и др.] // arXiv preprint arXiv:2008.12348. — 2020.

18. *Баймурзина, Д. Р.* Распознавание интентов с помощью нейросетей / Д. Р. Баймурзина // Тезисы конференции «Ломоносов — 2018». — 2018. — С. 183—185.

19. Deeppavlov: Open-source library for dialogue systems / M. Burtsev [и др.] // Proceedings of ACL 2018, System Demonstrations. — 2018. — С. 122—127.

20. *Baymurzina, D.* Language model embeddings improve sentiment analysis in Russian / D. Baymurzina, D. Kuznetsov, M. Burtsev // Komp'juternaja Lingvistika i Intellektual'nye Tehnologii. — 2019. — С. 53—62.

21. DREAM technical report for the Alexa Prize 2019 / Y. Kuratov [и др.] // Alexa Prize Proceedings. — 2020. — URL: https://d7qzviu3xw2xc.cloudfront.net/alexa/alexaprize/assets/pdf/sgc3/Moscow-DREAM.pdf.

22. Диалоговая система DREAM в конкурсе Alexa Prize Challenge 2019 / Ю. М. Куратов [и др.] // Труды МФТИ. — 2021. — Т. 13, № 3. — С. 62—89.

23. Evaluation of Conversational Skills for Commonsense / D. Baymurzina [и др.] // Proceedings of Dialog 2021. — 2021.

24. DREAM technical report for the Alexa Prize 4 / D. Baymurzina [и др.] // Alexa Prize Proceedings. — 2021. — URL: https://d7qzviu3xw2xc.cloudfront.net/alexa/alexaprize/docs/sgc4/MIPT-DREAM.pdf.

25. Программа разговорного навыка для проведения диалога о кино : а. с. / Д. Баймурзина, Д. Кузнецов (Российская Федерация) ; федеральное государственное автономное образовательное учреждение высшего образования «Московский физико-технический институт (национальный исследовательский университет)». — № 2021664221 ; заявл. 2021-08-25 ; опубл. 01.09.2021 (Российская Федерация). — 1 с.

26. Программа разговорных навыков, интегрирующих модели предсказания аспектов здравого смысла в диалоге : а. с. / Д. Баймурзина [и др.] (Российская Федерация) ; федеральное государственное автономное образовательное учреждение высшего образования «Московский физико-технический институт (национальный исследовательский университет)». — № 2021662601 ; заявл. 2021-07-23 ; опубл. 02.08.2021 (Российская Федерация). — 1 с.

27. Программа выбора финального ответа из реплик-кандидатов : а. с. / Д. Баймурзина, Ю. Куратов, М. Бурцев (Российская Федерация) ; федеральное государственное автономное образовательное учреждение высшего образования «Московский физико-технический институт (национальный исследовательский университет)». — № 2021662460 ; заявл. 2021-07-23 ; опубл. 29.07.2021 (Российская Федерация). — 1 с.

28. Среда для создания сценарных разговорных агентов : а. с. / Д. Кузнецов, Д. Баймурзина (Российская Федерация) ; федеральное государственное автономное образовательное учреждение высшего образования «Московский физико-технический институт (национальный исследовательский уни-

верситет)». — № 2021664168 ; заявл. 2021-08-25 ; опубл. 01.09.2021 (Российская Федерация). — 1 с.

29. *Colby, K. M.* Artificial paranoia / K. M. Colby, S. Weber, F. D. Hilf // Artificial Intelligence. — 1971. — Т. 2, № 1. — С. 1—25.

30. Turing-like indistinguishability tests for the validation of a computer simulation of paranoid processes / K. M. Colby [и др.] // Artificial Intelligence. — 1972. — Т. 3. — С. 199—221.

31. GUS, a frame-driven dialog system / D. G. Bobrow [и др.] // Artificial intelligence. — 1977. — Т. 8, № 2. — С. 155—173.

32. *Bellegarda, J. R.* Natural language technology in mobile devices: Two grounding frameworks / J. R. Bellegarda // Mobile Speech and Advanced Natural Language Solutions. — 2013. — С. 185—196.

33. *Яндекс.* Представляем голосового помощника Алису / Яндекс. — 2017. — URL: https://yandex.ru/blog/company/alisa.

34. The first conversational intelligence challenge / M. Burtsev [и др.] // The NIPS'17 Competition: Building Intelligent Systems. — Springer, Cham, 2018. — С. 25—46.

35. The second conversational intelligence challenge (convai2) / E. Dinan [и др.] // arXiv preprint arXiv:1902.00098. — 2019.

36. *Young, S. J.* Probabilistic methods in spoken–dialogue systems / S. J. Young // Philosophical Transactions of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences. — 2000. — Т. 358, № 1769. — С. 1389—1402.

37. *Hunt, A. J.* Unit selection in a concatenative speech synthesis system using a large speech database / A. J. Hunt, A. W. Black // 1996 IEEE International Conference on Acoustics, Speech, and Signal Processing Conference Proceedings. Т. 1. — IEEE. 1996. — С. 373—376.

38. Wavenet: A generative model for raw audio / A. v. d. Oord [и др.] // arXiv preprint arXiv:1609.03499. — 2016.

39. *Loper, E.* Nltk: The natural language toolkit / E. Loper, S. Bird // arXiv preprint cs/0205028. — 2002.

40. Enriching word vectors with subword information / P. Bojanowski [и др.] // Transactions of the Association for Computational Linguistics. — 2017. — Т. 5. — С. 135—146.

41. *Řehůřek, R.* Software Framework for Topic Modelling with Large Corpora / R. Řehůřek, P. Sojka. — 2010. — Май.

42. The Stanford CoreNLP natural language processing toolkit / C. D. Manning [и др.] // Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations. — 2014. — С. 55—60.

43. Tensorflow: A system for large-scale machine learning / M. Abadi [и др.] // 12th {USENIX} symposium on operating systems design and implementation ({OSDI} 16). — 2016. — С. 265—283.

44. *Ketkar, N.* Introduction to keras / N. Ketkar // Deep learning with Python. — Springer, 2017. — С. 97—111.

45. Pytorch: An imperative style, high-performance deep learning library / A. Paszke [и др.] // Advances in neural information processing systems. — 2019. — Т. 32. — С. 8026—8037.

46. Huggingface's transformers: State-of-the-art natural language processing / T. Wolf [и др.] // arXiv preprint arXiv:1910.03771. — 2019.

47. Rasa: Open source language understanding and dialogue management / T. Bocklisch [и др.] // arXiv preprint arXiv:1712.05181. — 2017.

48. ConveRT: Efficient and Accurate Conversational Representations from Transformers / M. Henderson [и др.] // arXiv preprint arXiv:1911.03688. — 2019.

49. Transfertransfo: A transfer learning approach for neural network based conversational agents / T. Wolf [и др.] // arXiv preprint arXiv:1901.08149. — 2019.

50. Recipes for building an open-domain chatbot / S. Roller [и др.]. — 2020. — arXiv: 2004.13637 [cs.CL].

51. Policy-Driven Neural Response Generation for Knowledge-Grounded Dialogue Systems / B. Hedayatnia [и др.] // arXiv preprint arXiv:2005.12529. — 2020.

52. *Finch, J. D.* Emora STDM: A Versatile Framework for Innovative Dialogue System Development / J. D. Finch, J. D. Choi // arXiv preprint arXiv:2006.06143. — 2020.

53. Distributed representations of words and phrases and their compositionality / T. Mikolov [и др.] // Advances in neural information processing systems. — 2013. — С. 3111—3119.

54. Efficient estimation of word representations in vector space / T. Mikolov [и др.] // arXiv preprint arXiv:1301.3781. — 2013.

55. *Pennington, J.* Glove: Global vectors for word representation / J. Pennington, R. Socher, C. Manning // Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). — 2014. — С. 1532—1543.

56. Attention is all you need / A. Vaswani [и др.] // arXiv preprint arXiv:1706.03762. — 2017.

57. Improving language understanding by generative pre-training / A. Radford [и др.] // URL https://s3-us-west-2. amazonaws. com/openai-assets/research-covers/languageunsupervised/language understanding paper. pdf. — 2018.

58. Language models are few-shot learners / T. B. Brown [и др.] // arXiv preprint arXiv:2005.14165. — 2020.

59. Xlnet: Generalized autoregressive pretraining for language understanding / Z. Yang [и др.] // Advances in neural information processing systems. — 2019. — Т. 32.

60. Roberta: A robustly optimized bert pretraining approach / Y. Liu [и др.] // arXiv preprint arXiv:1907.11692. — 2019.

61. Albert: A lite bert for self-supervised learning of language representations / Z. Lan [и др.] // arXiv preprint arXiv:1909.11942. — 2019.

62. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension / M. Lewis [и др.] // arXiv preprint arXiv:1910.13461. — 2019.

63. Electra: Pre-training text encoders as discriminators rather than generators / K. Clark [и др.] // arXiv preprint arXiv:2003.10555. — 2020.

64. Alquist 3.0: Alexa prize bot using conversational knowledge graph / J. Pichl [и др.] // arXiv preprint arXiv:2011.03261. — 2020.

65. Neural, Neural Everywhere: Controlled Generation Meets Scaffolded, Structured Dialogue / E. A. Chi [и др.] // Alexa Prize Proceedings. — 2021. — URL: https://developer.amazon.com/alexaprize/challenges/current-challenge/sgc4-proceedings.

66. SentiRuEval: testing object-oriented sentiment analysis systems in Russian / N. Loukachevitch [и др.] // Proceedings of International Conference Dialog. Т. 2. — 2015. — С. 3—13.

67. *Le, H. T.* Do convolutional networks need to be deep for text classification? / H. T. Le, C. Cerisara, A. Denis // Workshops at the Thirty-Second AAAI Conference on Artificial Intelligence. — 2018.

68. Supervised learning of universal sentence representations from natural language inference data / A. Conneau [и др.] // arXiv preprint arXiv:1705.02364. — 2017.

69. Application of a Hybrid Bi-LSTM-CRF model to the task of Russian Named Entity Recognition / M. Y. Arkhipov, M. S. Burtsev [и др.] // Conference on Artificial Intelligence and Natural Language. — Springer. 2017. — С. 91—103.

70. *Schuster, M.* Bidirectional recurrent neural networks / M. Schuster, K. K. Paliwal // IEEE transactions on Signal Processing. — 1997. — Т. 45, № 11. — С. 2673—2681.

71. Deep contextualized word representations / M. E. Peters [и др.] // arXiv preprint arXiv:1802.05365. — 2018.

72. RuSentiment: An Enriched Sentiment Analysis Dataset for Social Media in Russian / A. Rogers [и др.] // Proceedings of the 27th International Conference on Computational Linguistics. — 2018. — С. 755—763.

73. Exploring the limits of language modeling / R. Jozefowicz [и др.] // arXiv preprint arXiv:1602.02410. — 2016.

74. Character-aware neural language models / Y. Kim [и др.] // Thirtieth AAAI Conference on Artificial Intelligence. — 2016.

75. Semi-supervised sequence tagging with bidirectional language models / M. E. Peters [и др.] // arXiv preprint arXiv:1705.00108. — 2017.

76. Learned in translation: Contextualized word vectors / B. McCann [и др.] // Advances in Neural Information Processing Systems. — 2017. — С. 6294—6305.

77. *Robertson, S.* Understanding inverse document frequency: on theoretical arguments for IDF / S. Robertson // Journal of documentation. — 2004. — Т. 60, № 5. — С. 503—520.

78. *Kim, Y.* Convolutional neural networks for sentence classification / Y. Kim // arXiv preprint arXiv:1408.5882. — 2014.

79. Learning phrase representations using RNN encoder-decoder for statistical machine translation / K. Cho [и др.] // arXiv preprint arXiv:1406.1078. — 2014.

80. *Johnson, R.* Supervised and semi-supervised text categorization using LSTM for region embeddings / R. Johnson, T. Zhang // arXiv preprint arXiv:1602.02373. — 2016.

81. *Рубцова, Ю.* Автоматическое построение и анализ корпуса коротких текстов (постов микроблогов) для задачи разработки и тренировки тонового классификатора / Ю. Рубцова // Инженерия знаний и технологии семантического веба. — 2012. — Т. 1. — С. 109—116.

82. Recursive deep models for semantic compositionality over a sentiment treebank / R. Socher [и др.] // Proceedings of the 2013 conference on empirical methods in natural language processing. — 2013. — С. 1631—1642.

83. Scenariosa: A large scale conversational database for interactive sentiment analysis / Y. Zhang [и др.] // arXiv preprint arXiv:1907.05562. — 2019.

84. Universal Sentence Encoder / D. M. Cer [и др.] // ArXiv. — 2018. — Т. abs/1803.11175.

85. Towards Coherent and Engaging Spoken Dialog Response Generation Using Automatic Conversation Evaluators / S. Yi [и др.] // Proceedings of the 12th International Conference on Natural Language Generation. — Tokyo, Japan : Association for Computational Linguistics, 10–11.2019. — С. 65—75. — URL: https://www.aclweb.org/anthology/W19-8608.

86. *Yu, D.* Midas: A dialog act annotation scheme for open domain human machine spoken conversations / D. Yu, Z. Yu // arXiv preprint arXiv:1908.10023. — 2019.

87. Evaluating commonsense in pre-trained language models / X. Zhou [и др.] // Proceedings of the AAAI Conference on Artificial Intelligence. Т. 34. — 2020. — С. 9733—9740.

88. WINOGRANDE: An Adversarial Winograd Schema Challenge at Scale / K. Sakaguchi [и др.] // ArXiv. — 2019. — Т. abs/1907.10641.

89. Abductive Commonsense Reasoning / C. Bhagavatula [и др.] // International Conference on Learning Representations. — 2020. — URL: https : / / openreview.net/forum?id=Byg1v1HKDB.

90. A survey of commonsense knowledge acquisition / L.-J. Zang [и др.] // Journal of Computer Science and Technology. — 2013. — Т. 28, № 4. — С. 689—719.

91. COMET: Commonsense Transformers for Automatic Knowledge Graph Construction / A. Bosselut [и др.] // Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. — Florence, Italy : Association for Computational Linguistics, 07.2019. — С. 4762—4779. — URL: https://www.aclweb.org/anthology/P19-1470.

92. Atomic: An atlas of machine commonsense for if-then reasoning / M. Sap [и др.] // Proceedings of the AAAI Conference on Artificial Intelligence. Т. 33. — 2019. — С. 3027—3035.

93. *Speer, R.* Conceptnet 5.5: An open multilingual graph of general knowledge / R. Speer, J. Chin, C. Havasi // Thirty-First AAAI Conference on Artificial Intelligence. — 2017.

94. RoBERTa: A robustly optimized BERT pretraining approach. arXiv 2019 / Y. Liu [и др.] // arXiv preprint arXiv:1907.11692. —.

95. *Williams, A.* A broad-coverage challenge corpus for sentence understanding through inference / A. Williams, N. Nangia, S. R. Bowman // arXiv preprint arXiv:1704.05426. — 2017.

96. A large annotated corpus for learning natural language inference / S. R. Bowman [и др.] // Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP). — Association for Computational Linguistics, 2015.

97. Comet-atomic 2020: On symbolic and neural commonsense knowledge graphs / J. D. Hwang [и др.] // arXiv preprint arXiv:2010.05953. — 2020.

98. Like hiking? You probably enjoy nature: Persona-grounded Dialog with Commonsense Expansions / B. P. Majumder [и др.] // arXiv preprint arXiv:2010.03205. — 2020.

99. Further Advances in Open Domain Dialog Systems in the Fourth Alexa Prize Socialbot Grand Challenge / S. Hu, Y. Liu, A. Gottardi [и др.] // Alexa Prize Proceedings. — 2021.

100. Transomcs: From linguistic graphs to commonsense knowledge / H. Zhang [и др.] // arXiv preprint arXiv:2005.00206. — 2020.

101. Commonsense-Focused Dialogues for Response Generation: An Empirical Study / P. Zhou [и др.] // arXiv preprint arXiv:2109.06427. — 2021.

102. A decomposable attention model for natural language inference / A. P. Parikh [и др.] // arXiv preprint arXiv:1606.01933. — 2016.

103. LightGBM: A Highly Efficient Gradient Boosting Decision Tree / G. Ke [и др.] // Advances in Neural Information Processing Systems 30 / под ред. I. Guyon [и др.]. — Curran Associates, Inc., 2017. — С. 3146—3154. — URL: http://papers.nips.cc/paper/6907-lightgbm-a-highly-efficient-gradient-boosting-decision-tree.pdf.

104. Learning to rank question answer pairs with holographic dual lstm architecture / Y. Tay [и др.] // Proceedings of the 40th international ACM SIGIR conference on research and development in information retrieval. — 2017. — С. 695—704.

105. Sequential matching network: A new architecture for multi-turn response selection in retrieval-based chatbots / Y. Wu [и др.] // arXiv preprint arXiv:1612.01627. — 2016.

106. Multi-turn response selection for chatbots with deep attention matching network / X. Zhou [и др.] // Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). — 2018. — С. 1118—1127.

107. Do Response Selection Models Really Know What's Next? Utterance Manipulation Strategies for Multi-turn Response Selection / T. Whang [и др.] // arXiv preprint arXiv:2009.04703. — 2020.

108. Improving Dialogue Breakdown Detection with Semi-Supervised Learning / N. Ng [и др.] // arXiv preprint arXiv:2011.00136. — 2020.

109. *Chen, T.* Xgboost: A scalable tree boosting system / T. Chen, C. Guestrin // Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining. — 2016. — С. 785—794.

110. *Dorogush, A. V.* CatBoost: gradient boosting with categorical features support / A. V. Dorogush, V. Ershov, A. Gulin // arXiv preprint arXiv:1810.11363. — 2018.

111. *Dawid, A. P.* Maximum likelihood estimation of observer error-rates using the EM algorithm / A. P. Dawid, A. M. Skene // Journal of the Royal Statistical Society: Series C (Applied Statistics). — 1979. — Т. 28, № 1. — С. 20—28.

112. Gunrock: Building a human-like social bot by leveraging large scale real user data / C.-Y. Chen [и др.] // 2nd Proceedings of Alexa Prize (Alexa Prize 2018). — 2018.

Список рисунков

Список таблиц