

На правах рукописи



Юсупов Идрис Фаргатович

Контекстный диалоговый агент

Специальность 05.13.17 —
«Теоретические основы информатики»

Автореферат
диссертации на соискание учёной степени
кандидата технических наук

Москва — 2020

Работа прошла апробацию в Федеральном государственном автономном образовательном учреждении высшего образования «Московский физико-технический институт (национальный исследовательский университет)».

Научный руководитель: кандидат физико-математических наук
Бурцев Михаил Сергеевич

Ведущая организация: Федеральное государственное бюджетное учреждение науки Институт проблем передачи информации им. А.А. Харкевича Российской академии наук (ИППИ РАН)

Защита состоится 25 декабря 2020 г. в 11 часов на заседании диссертационного совета ФПМИ.05.13.17.004 по адресу: 141701, Московская область, г. Долгопрудный, Институтский переулок, д.9.

С диссертацией можно ознакомиться в библиотеке и на сайте Московского физико-технического института (национального исследовательского университета) <https://mipt.ru/education/post-graduate/soiskateli-tekhnicheskie-nauki.php>.

Работа представлена 13 октября 2020 г. в Аттестационную комиссию федерального государственного автономного образовательного учреждения высшего образования «Московский физико-технический институт (национальный исследовательский университет)» для рассмотрения советом по защите диссертаций на соискание ученой степени кандидата наук, доктора наук в соответствии с п. 3.1 ст. 4 Федерального закона «О науке и государственной научно-технической политике».

Общая характеристика работы

Актуальность темы. Сейчас существует множество программ для обмена сообщениями, которыми пользуются во всем мире. Например, база пользователей программы Telegram составляет более 400 миллионов активных пользователей по состоянию на апрель 2020 года. Компании повсюду используют этот канал для коммуникации с клиентами. Стоит задача автоматизации такой коммуникации, т.к. запросы пользователей часто носят однотипный характер. На сегодняшний день популярны голосовые ассистенты, например, Amazon Alexa, Apple Siri, Яндекс Алиса. С ними человек может общаться на свободные темы или для решения повседневных задач, таких как заказ такси, бронирования столика в ресторане и т.п.

Кроме коммерческого интереса, в последние годы было организовано множество соревнований по созданию диалоговых агентов. В 2017 году впервые был проведен Alexa Prize Challenge - соревнование (контест), целью которого является разработка диалогового агента (чат-бота, бота) способного вести диалог с человеком на разные темы. Разработка такого диалогового агента требует использования знаний разных дисциплин, таких как компьютерная лингвистика, программная инженерия, искусственный интеллект. С 1991 года проводится AI Loebner - ежегодный контест, в котором необходимо разработать диалогового агента, способного пройти Тест Тьюринга. В 2017 году прошел первый ConyAI - The Conversational Intelligence Challenge, где требовалось разработать чат-бота ведущего диалог о заданном параграфе текста с человеком. В 2018 году прошло второе такое соревнование, где в отличие от первого, боту присуждалось случайное описание его личности.

Несмотря на весь прогресс, AI Loebner продолжается, т.е. Тест Тьюринга не пройден, лучший диалоговый агент Alexa Prize 2019 показывает оценку 3.8 из 5 по результатам финальной оценки судей. Полной автоматизации клиентской поддержки добиться не удастся, а голосовые ассистенты часто ошибаются и разочаровывают конечных пользователей.

На текущий момент программный код практически каждого диалогового агента уникален. Например, ни один из участников Alexa Prize не выкладывал исходный код в открытый доступ. Находящиеся же диалоговые агенты с открытым исходным кодом плохо документированы и тяжело расширяемы. Наличие гибкого, универсального и открытого программного фреймворка для создания диалоговых агентов, позволило бы разработчикам “стоять на плечах гигантов”, а не начинать каждый раз разработку с нуля.

Одной из основных задач в диалоговых агентах является задача построения диалогового менеджера. Диалоговый менеджер отвечает за то, что бот должен делать на текущем шаге в диалоге. Обычно в начале разработки отсутствуют данные о диалогах из-за чего возникает проблема холодного старта. Это замедляет и увеличивает стоимость разработки бота, из-за необходимости размечать данные и использовать эвристики.

Анализ переходов между репликами является первой ступенью в создании связанного диалога. По нему можно понять какой переход был удачным, а какой нет и скорректировать его. Разнообразие контекстов в диалогах, которое возникает при общении человека и бота огромное, поэтому полезно представление диалоговых данных в виде графа. Вершина графа - семантически значимое представление контекстов, например, “Приветствие”, “Знакомство” и т.п. Ребро - переход между вершинами. Имея алгоритм построения такого графа можно эффективнее анализировать большие диалоговые корпуса, а в перспективе автоматически строить логику диалогового менеджера.

Целью данной работы является разработка диалогового агента способного вести диалог с человеком в рамках определенного контекста, а также метода построения и использования диалогового графа для визуализации и улучшения качества диалога.

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

1. Обзор современных методов построения диалоговых агентов.
2. Разработка и исследование архитектур диалоговых агентов.
3. Создание и исследование диалогового менеджера с решением проблемы холодного старта.
4. Разработка программного комплекса диалогового агента.
5. Оценка качества разработанного диалогового агента.
6. Анализ диалогов проведенных диалоговым агентом.
7. Исследование методов построения и использования диалоговых графов в диалоговой системе.

Научная новизна:

1. Разработана и исследована эффективная и доказавшая свою состоятельность в двух соревнованиях, архитектура диалогового агента, способного вести разговор в определенном контексте.
2. Разработан и исследован диалоговый менеджер, решающий проблему холодного старта, использующий методы обучения с учителем.
3. Разработан и исследован метод построения диалогового графа из диалогового корпуса, который является инструментом визуализации и источником признаков, которые улучшают качество на целевой задаче.
4. Созданы и апробированы комплексы программ, включающих в себя как диалогового агента в целом, так и его отдельные компоненты.

Практическая значимость Разработанный диалоговый агент показал практическую значимость заняв 1 место на соревновании ConvAI 1 и войдя в тройку лучших на этапе хакатона в ConvAI 2. Также идеи предложенные в нем получили дальнейшее развитие в Alexa Prize 2019. Компонент по генерации вопросов был использован в коммерческом проекте Autofaq.ai. Программный

комплекс диалогового агента документирован и находится в открытом доступе под свободной лицензией.

Методология и методы исследования. В диссертации применялись методы компьютерной лингвистики, машинного обучения, искусственного интеллекта, программной инженерии.

Основные положения, выносимые на защиту:

1. Разработан диалоговый агент ведущий разговор в определенном контексте. Показана способность поддерживать как контекст в смысле обсуждения определенного текста, так и контекст в смысле описания персоналии агента.
2. Разработан адаптированный под холодный старт диалоговый менеджер. Для его работы требуются только данные самих навыков, либо несколько подготовленных фраз.
3. Предложен метод построения диалогового графа из диалогового корпуса, который является полезным инструментом визуализации. Предложенный метод применим для корпусов разных размеров и тематик. Диалоговый граф также является источником полезных признаков, улучшающие качество на целевой задаче.

Достоверность полученных результатов обеспечивается экспериментами на наборах диалоговых данных, а также участием в соревнованиях ConvAI 1, ConvAI 2, Alexa Prize 2019.

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

- The Conversational Intelligence Challenge section on NIPS 2017 Competition Track Workshop, международная конференция NIPS, Long Beach, 2017.
- Международная конференция COLING, Santa Fe, 2018.
- Mini-Workshop: Stochastic Processes and Probabilistic Models in Machine Learning, семинар ВШЭ, Москва, 2018.
- AMC Makeathon 2019: Big data for social event benefit, Москва, 2019.

Публикации. Основные результаты по теме диссертации изложены в 4 печатных изданиях, 1 из которых изданы в журналах, рекомендованных ВАК, 1 — в тезисах докладов. [A1] - свидетельство о государственной регистрации программы для ЭВМ.

Благодарности. Автор выражает глубокую признательность своему научному руководителю, кандидату физико-математических наук, Бурцеву Михаилу Сергеевичу за ценные замечания и неоценимую помощь в подготовке диссертации. Автор благодарит Юрия Куратова и Дмитрия Карпова за идеи и помощь. Также автор благодарен компании AutoFAQ.ai за финансовую поддержку и Элине Юсуповой за редактирование публикаций.

Содержание работы

Во **введении** обосновывается актуальность диссертационной работы, сформулированы цели и задачи, аргументирована научная новизна, представлены результаты и положения, выносимые на защиту, приведена краткая структура диссертации по главам.

Первая глава посвящена описанию области диалоговых агентов.

В **разделе 1.1** дается определение диалогового агента и его архитектуры.

Диалоговый агент (диалоговая система, агент, чат-бот, бот) - это программа, которая ведет разговор с человеком через текстовый или звуковой интерфейс. Агенты разделяются на целеориентированных (task-oriented), предназначенных для решения определенной задачи, например бронирование номера в гостинице или заказа такси, и общего назначения (general purpose). Агенты общего назначения должны уметь вести диалог на общие темы. Также диалоговых агентов разделяют по области действия (домену). Боты с открытым доменом покрывают широкий спектр тем диалога, тогда как чат-бот с закрытым доменом специализируется только на конкретных темах.

Диалоговые агенты имеют сложную архитектуру и состоят из нескольких модулей (рис. 1):

- понимания естественного языка (natural language understanding),
- диалогового менеджера (dialogue management),
- генерации естественного языка (natural language understanding).

Также могут присутствовать модули распознавания речи и синтеза речи. Далее рассмотрим три перечисленных выше модуля.

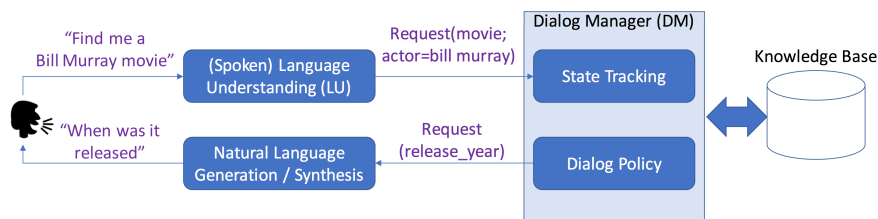


Рисунок 1 — Архитектура диалогового агента. Рисунок взят из [1].

Модуль понимания естественного языка решает задачу обработки входящего сырого текстового запроса пользователя и его отображения в структурное представление. Типичные задачи, которые решает этот модуль включает в себя фильтрацию входящего запроса пользователя, его нормализацию, извлечение именованных сущностей и фактов, классификацию запроса по темам.

Модуль диалогового менеджера является ядром диалоговой системы. На основе информации полученной из модуля понимания естественного языка происходит выбор действия, который должен совершить бот.

В [2; 3] основные задачи диалогового менеджера определяются как:

- обновление диалогового контекста на основании разбора текущего сообщения;
- выдача контекстно-зависимых разборов сигналов;
- координирование поведения в диалоге, принятие решения о том, каково будет содержание следующей реплики и когда ее следует выдать.

Модуль генерации естественного языка выдает реплику бота на основе информации полученной от диалогового менеджера. Например, диалоговый менеджер возвращает действие “приветствие”, соответственно модуль генерирует реплику для приветствия.

В **разделе 1.2** описываются различные способы оценки качества диалоговых агентов.

Оценку качества диалогового агента проводят двумя способами: автоматическим и ручным, с помощью человека. Для автоматической оценки качества требуется размеченный диалоговый корпус.

Автоматический способ оценки применим для целеориентированных диалоговых агентов или для агентов с ограниченным доменом, которые не развиваются с течением времени. Для реализации целеориентированных систем, можно использовать фрейм, который состоит из слотов с параметрами ситуации и значений слотов.

Если диалоговая система не использует фреймы, а оперирует только репликами, то качество измеряют мерами из области информационного поиска, например Recall@k, которая показывает долю релевантных реплик среди k лучших, выбранных моделью. Также часто используют метрику BLEU [4], применяемую в машинном переводе, которая учитывает сколько n-грамм совпадает в переводе и в эталоне, а затем по определенной формуле выводит оценку качества по шкале от 0 до 100 [3]. В недавних работах было предложено оценивать качество диалоговых агентов путем обучения модели, предсказывающую такую оценку, используя размеченную выборку [5; 6].

В диалоговых агентах общего назначения с открытым доменом автоматические метрики не могут обеспечить достаточной полноты, т.к. человек может общаться с ними о чем угодно. Поэтому здесь необходимо использовать ручную оценку качества. Ручная оценка подразумевает проведение диалога бота с человеком, где человек в конце диалога ставит оценку, обычно по пятибалльной шкале, где 1 - диалог с ботом совсем не понравился, 5 - очень понравился. Получение состоятельной оценки требует большого количества диалогов, что является материально и временно затратным действием. Для научных групп здесь на передний план выходят соревнования (контесты) по разработке диалоговых агентов, т.к. участие в них дает возможность бесплатно получить большое количество оценок и оценить качество в сравнении с другими участниками соревнования.

В 2017 году был организован Alexa Prize Challenge целью которого является создание диалогового агента общего назначения для ведения диалога в

открытом домене [7]. В отличие от предыдущего соревнования здесь не ставится цель пройти Тест Тьюринга. Главной задачей бота является ведение диалога длительностью 20 минут и средней оценкой пользователей 4 из 5. За решение такой задачи победившей команде присуждается приз в 1 млн. долларов. Соревнование также стало ежегодным.

Также в 2017 впервые было проведено соревнование ConvAI - The First Conversational Intelligence Challenge [8]. Задачей разработчиков чатботов было создание бота, способного вести диалог о случайном абзаце статьи из Википедии. Победители достигли оценки в 2.78 из 5, верхней границей можно считать оценку человека в 3.8. В 2018 году соревнование провели во второй раз, но изменили задачу [9]. Теперь боту давалась определенная персона и бот должен был говорить о себе в ее контексте. В ней кроме пятибалльной пользовательской оценки, также были использованы автоматические метрики, такие как Perplexity, Accuracy@1, F1-мера.

В разделе 1.3 описаны текущие проблемы диалоговых агентов.

Первой проблемой можно назвать отсутствие хорошо документированных, простых в использовании, расширяемых программных комплексов для построения диалоговых агентов. Еще ни один участник Alexa Prize не выкладывал программный код своих ботов в открытый доступ. Код победителя последних AI Loebner 2017-2019 - Mitsuku¹, также закрыт. Стоит отметить, что код победителя AI Loebner 2004 - A.L.I.C.E [10] есть в открытом доступе и его активно используют, как один из подмодулей в диалоговых агентах на соревнованиях. По условиям соревнований ConvAI 1 и 2 необходимо выкладывать код в открытый доступ, что позволяет использовать их как один из подмодулей в своем агенте. Но хотелось бы иметь библиотеку для построения диалогового агента, наподобие Django² и Rails³ для веб-приложений. Сейчас каждый диалоговый агент в смысле программного решения уникален. Если бы разработчики использовали единый программный фреймворк - это бы позволило другим командам ускорить скорость разработки, проще реализовывать и поддерживать сложную логику, увеличить сопровождаемость и надежность решений, легче находить и вводить новых разработчиков.

В начале разработки диалоговой системы отсутствуют данные о диалогах из-за чего возникает проблема “холодного старта”. Это замедляет и увеличивает стоимость разработки бота, т.к. нужно размечать данные, либо использовать эвристики до накопления данных. Это особенно актуально для диалогового менеджера, т.к. в качестве подмодулей для генерации и понимания естественного языка можно использовать те, что находятся в открытом доступе. Диалоговый менеджер - уникальный у каждого диалогового агента. Так например в Mitsuku

¹<https://www.pandorabots.com/mitsuku/>

²<https://www.djangoproject.com>

³<https://rubyonrails.org>

написано множество правил, а один из участников Alexa Prize 2017 MILABOT [11] собирал размеченные данные через Amazon Mechanical Turk⁴.

Изменения в диалоговом агенте надо делать анализируя поступающие диалоги пользователей. Таких диалогов может быть тысячи в день. Можно случайно отбирать диалоги, читать их и фиксировать задачи на улучшение. Также важным является анализ различных статистик, например как часто был вызван тот или иной навык в диалоговой системе. Анализ переходов между навыками является первой ступенью в создании связанного диалога. По нему будет ясно какой переход был удачным, а какой нет и скорректировать его. Разнообразие контекстов в диалогах, которые возникают при общении человека и бота огромное, поэтому полезно представление диалоговых данных в виде графа. Вершиной графа может быть номер хода в диалоге, а ребром переход на следующий номер хода. Один ход может быть представлен несколькими вершинами, при условии, что вершина получила семантически значимое имя. Имея алгоритм построения такого графа получится более эффективно анализировать большие диалоговые корпуса, а в перспективе, автоматически строить логику диалогового менеджера.

Вторая глава посвящена описанию разработанного в рамках соревнования NIPS Conversational Intelligence Challenge (ConvAI 1) диалогового агента bot#1337, его участию в соревновании ConvAI 2 и механизму переходов между навыками в диалоговом агенте DREAM для соревнования Alexa Prize 2019.

В **разделе 2.1** представлен обзор соревнования и связанные работы.

В NIPS Conversational Intelligence Challenge 2017 [12] требовалось разработать диалоговую систему, которая обсуждает с пользователем короткий отрывок текста из Википедии. Каждый диалог в конце оценивался человеком по шкале от 1 (плохо) до 5 (отлично) по 3-м показателям:

- общему качеству диалога,
- глубине обсуждения предложенного текста диалоговой системой,
- вовлеченности диалогового агента в беседу.

Каждую реплику бота можно было пометить как подходящую или наоборот. Человек оценивал системы вслепую, не зная ее разработчиков.

В **разделе 2.2** дается детальное описание разработанного диалогового агента, победителя соревнования ConvAI 1 bot#1337, его навыков и диалогового менеджера.

Разработанный диалоговый агент состоит из двух основных компонент - диалогового менеджера и навыков. Навык - это модель, которая принимает на вход текст и диалоговый контекст и возвращает реплику ответа. Диалоговый менеджер решает какой навык использовать на текущем шаге диалога.

Верхнеуровневая архитектура разработанной диалоговой системы представлена на рис. 2. На вход системе подается параграф текста о котором идет беседа, контекст диалога и текущая реплика на которую необходимо дать ответ. Затем диалоговый менеджер по входным данным решает какой навык выбрать. После этого выбранный навык используя входную информацию выдает ответ.

⁴<https://www.mturk.com>

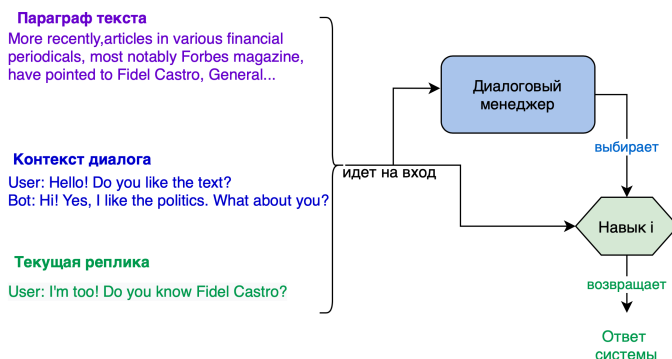


Рисунок 2 — Верхнеуровневая архитектура диалоговой системы bot#1337.

Для ведения диалога о тексте требуются различные навыки. Во-первых, бот должен уметь приветствовать пользователя, отвечать на вопросы о тексте и задавать их. Во-вторых, заданный текст бывает длинным, поэтому важно уметь делать автоматическое реферирование текста. Часто пользователь может иметь желание поговорить с ботом на общие темы.

Разговорные навыки по OpenSubtitles и Facebook новостям используют нейросетевые методы машинного перевода и основаны на моделях типа последовательность-последовательность (sequence-to-sequence) с механизмом внимания. Они обучены с помощью открытого программного комплекса OpenNMT. OpenNMT - это библиотека для реализации моделей глубокого обучения, в особенности, моделей типа последовательность-последовательность. Она покрывает задачи машинного перевода, реферирования, отображения изображения в текст и распознавания голоса. Библиотека также может решать задачи построения языковых моделей и разметки последовательности [13].

Обучение осуществлялось на уровне представления слов. Размерность векторных представлений слов была равна 150, веса слов обновлялись во время обучения. Для итеративного обновления весов модели на основе обучающей выборки был применен метод оптимизации ADAM (Adaptive Moment Estimation, метод Адаптивной Оценки Моментов) [14]. ADAM - это метод оптимизации, который можно использовать вместо классического стохастического градиентного спуска. Обновления весов для ADAM основано на использовании оценок двух моментов. Первый использует вычисленные ранее значения частных производных, а второй их квадраты. ADAM часто используют как метод по умолчанию, т.к. он считается устойчивым к выбору значений гиперпараметров.

Разговорный навык по OpenSubtitles принимает на вход контекст диалога и генерирует реплику на выходе. В качестве модели используется 2-х слойный энкодер-декодер с ячейкой LSTM размерности 2048. Модель обучалась на открытом корпусе субтитров OpenSubtitles [15].

Разговорный навык по Facebook новостям на вход принимает контекст диалога и текст, о котором идет беседа. Текст и контекст разделены специальным токеном “EOP” (end of paragraph - конец параграфа). Контекст диалога S конкатенировался с текстом $P = \{p_1, p_2, \dots, p_{N_p}\}$, состоящий из слов p_i : $S = P \oplus S$. На выходе также генерируется реплика. В качестве модели используется 1-слойный энкодер-декодер с ячейкой LSTM размерности 1024. За обучающую выборку выступает датасет новостей из Facebook⁵, который состоит из постов-новостей и комментариев из Facebook.

Разговорный навык Alice [10] принимает на вход контекст диалога и генерирует реплику на выходе. Навык построен с помощью AIML (Artificial Intelligence Markup Language) - хорошо документированного, широко используемого диалекта языка XML для декларативной разработки диалоговых агентов. В навыке используется открытая версия Alice из Github⁶, которая включает в себя большой набор грамматик, покрывающие разговоры на общие темы.

Разговорные навыки использующие модели типа последовательность-последовательность на выходе генерируют несколько реплик. Сгенерированные реплики могут быть короткими или содержать неприемлемые и повторяющиеся слова. Такие реплики проходят через алгоритм фильтрации (алг. 1). Если фильтрацию не прошла ни одна реплика, то запускается навык Alice или возвращается одна из запасных реплик. Запасные реплики включают в себя такие фразы как “Do you like this text?”, “What do you think about the competition?” и т.п.

Алгоритм 1: Фильтрация реплик-кандидатов

- 1 **Вход** реплики кандидаты **Выход** отфильтрованные реплики
 - 2 удалить дублирующие реплики;
 - 3 удалить короткие реплики;
 - 4 удалить реплики с большим количеством повторяющихся слов;
 - 5 удалить реплики с большим количеством стоп-слов;
 - 6 удалить реплики с нежелательными словами;
 - 7 вернуть отфильтрованные реплики;
-

Для навыка генерации вопроса была реализована модель типа последовательность - последовательность с вниманием, использующая дополнительные признаки [16]. В качестве модели используется 2-х слойный энкодер-декодер с двунаправленной ячейкой GRU размерности 512. Она обучена на датасете SQuAD [17] с помощью библиотеки OpenNMT. Обучение осуществлялось на уровне представления слов с методом оптимизации ADAM. Размерность векторных представлений слов была равна значению 300. Размерность лексических признаков и признака ответа была представлена векторами размерности 32. Этот навык принимает на вход параграф, именованные сущности и лексические

⁵<https://github.com/jbencina/facebook-news>

⁶<https://github.com/sld/convai-bot-1337/tree/master/ALICEChatAPI>

Алгоритм 2: Алгоритм проверки ответа на вопрос.

Вход: t : правильный ответ на вопрос

Вход: p : ответ на вопрос, который выдал пользователь

Вход: $d(t, p)$: расстояние Левенштейна между t и p

Вход: $r()$: функция, возвращающая случайное значение от 0 до 1

Если $d(t, p) \geq 0.95$ **То**

Вернуть Good! This is correct answer!

Иначе

Если $d(t, p) \geq 0.8$ **То**

Вернуть I think you mean: t ! My congratulations!

Иначе

Вернуть Hint: first 3 letters is t_{13} . Try again, please.

Конец Если

Конец Если

признаки, которые извлекались с помощью открытого программного комплекса Stanford CoreNLP [18], и на выходе возвращает сгенерированный вопрос. Во время стадии предсказания модель принимает на вход именованные сущности в качестве ответов, по которым должны быть выданы вопросы.

Навык проверки ответа на вопрос связан с предыдущим навыком. Правильный ответ для каждого вопроса является именованной сущностью, которая извлекается с помощью Stanford CoreNLP. Этот навык проверяет правильность ответа, который дал пользователь, используя расстояние Левенштейна и алг. 2. Впоследствии навык использует шаблоны, по которым выдается ответ пользователю, например, для ответа “google” следующая реплика будет выдана, в случае если пользователь ответил неверно, “You can do better! Hint: first 3 letters is goo”.

Навык ответа на вопрос использует модель BiDAF - bidirectional attention flow model [19] - двунаправленную рекуррентную нейронную сеть с механизмом внимания. На вход модели идет вопрос пользователя, текст и на выходе возвращается ответ. Впоследствии, этот ответ используется в одной из шаблонных фраз, которая выдается пользователю (например, “I think the answer is ...”). В данной работе использована открытая версия модели BiDAF размещенная на Github⁷.

Параграф текста, о котором надо беседовать, состоит из 5-10 предложений. *Навык реферирования* может помочь пользователю в экономии времени на чтение и в вовлечении его в беседу.

Для этого навыка была использована модель типа последовательность-последовательность с вниманием, предобученная на датасете Gigaword на OpenNMT⁸. В качестве модели используется 2-х слойный энкодер-декодер с ячейкой LSTM размерности 500. Размерность векторных представлений слов равна 500. Модель принимает на вход куски из заданного текста и возвращает

⁷<https://github.com/allenai/bi-att-flow>

⁸<http://opennmt.net/Models/#english-summarization>

сокращенные версии. Впоследствии, этот ответ используется в одной из шаблонных фраз, которая выдается пользователю (например, “Maybe this article’s main idea is ...”). Для выбора лучшего ответа применяется алг. 1.

Упоминание основной темы текста может вовлечь пользователя в беседу. *Навык определения темы* реализован с помощью библиотеки для тематического моделирования BigARTM. BigARTM - это инструмент для тематического моделирования, основанный на методе аддитивной регуляризации тематических моделей. Аддитивная регуляризации в BigARTM позволяет разрабатывать модели с заданными свойствами путём комбинирования готовых модулей-регуляризаторов [20].

Было реализовано 2 навыка не использующих контекст диалога, текущую реплику и параграф текста. Они оба состоят из фраз написанных вручную, которые выбираются случайно и отправляются пользователю.

Навык приветствия используется в начале диалога, если пользователь молчит в течение некоторого времени. Он включает в себя такие фразы как “Well hello there!”, “Hiya!”, и т.п.

Навык общих фраз используется во время диалога, если пользователь не проявляет активности в течение некоторого времени или если все другие навыки вернули пустой ответ. Он включает в себя такие фразы как “What do you think about ConvAI competition?”, “Do you like this text?”, и т.п.

Выбор навыка производится *диалоговым менеджером*. Задачей диалогового менеджера является выбор навыка с помощью которого будет выдан ответ пользователю. В данной работе описан диалоговый менеджер, который решает проблему холодного старта. Для его работы требуются только данные самих навыков, либо несколько подготовленных фраз.

Диалоговый менеджер, представленный на рис. 3, использует 2 классификатора, которые принимают на вход контекст диалога или реплику пользователя и возвращают навык, который должен выдать ответ.

Первый классификатор использует небольшую обучающую выборку, состоящую из нескольких фраз на навык, среднее значение векторного представления слов (на основе GloVe представлений⁹) и метод k-ближайших соседей (k-NN).

Второй классификатор использует большую обучающую выборку и библиотеку fastText. FastText - это библиотека для реализации классификаторов текстов и обучения представлений [21]. FastText модель была обучена¹⁰ предсказывать 5 классов (рис. 3).

На рис. 4 представлен линейный классификатор fastText. Для подсчета распределения вероятностей по классам используется функция softmax f . Обучения набора из N документов проводится путем минимизации функции

⁹<https://nlp.stanford.edu/projects/glove/>

¹⁰Скрипт обучения доступен по ссылке <https://github.com/sld/convai-bot-1337/tree/master/classifiers>

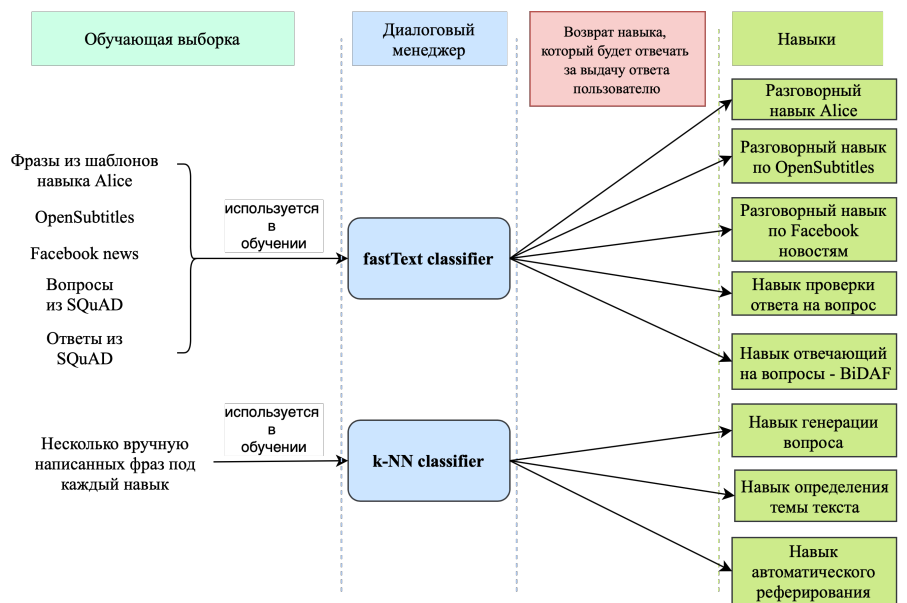


Рисунок 3 — Диалоговый менеджер с классификаторами, их обучающей выборкой и множеством наименований классов (навыков).

кросс-энтропии:

$$-\frac{1}{N} \sum_{n=1}^N y_n \log(f(BAx_n)),$$

где x_n - нормализованный мешок признаков для n -го документа, y_n - метка класса, A и B матрицы весов. Модель обучается параллельно методом стохастического градиентного спуска с линейно затухающей скоростью обучения.

Все навыки могут быть поделены на 2 категории: без обучающей выборки или с ней. Для разговорных и вопросно-ответных навыков имеются данные с репликами, вопросами и ответами. Их наличие позволяет обучить классификатор выбрать более подходящую входную реплику пользователя для какого-либо датасета (и следовательно навыка). Навыки реферирования, определения темы и генерации вопросов не имеют данных, по которым можно понять, что нужно их выбирать. Для этих навыков было написано по 3-10 фраз, которые были использованы с k-NN вместе со средним значением векторных представлений слов фразы.

Например, имеется 2 навыка - Open Subtitles и определение темы. Первый навык использует классификатор, обученный с помощью fastText, т.к. вместе с ним идет множество диалоговых реплик, таких как “They still behind us?”, “Senora, give me a break!” и т.п. Второй навык таких реплик не имеет. Написав

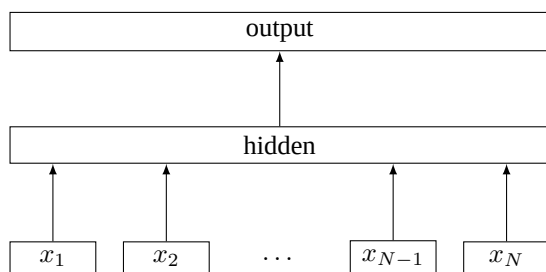


Рисунок 4 — Архитектура классификатора fastText с признаками n-грам $x_1, \dots, x_N$. Эти признаки кодируются в векторные представления и усредняются для получения скрытого состояния (hidden).

для него несколько фраз, таких как “What is the theme of text?”, “Say me theme”, “This text main topics”, можно обучить k-NN классификатор.

В разделе 2.3 приводится анализ работы диалогового агента, включая сравнение с другой победившей системой, анализ частоты использования навыков, оценку качества диалогового менеджера и сведения, полученные после чтения диалогов.

Первая версия bot#1337 включала в себя разговорный и вопросно-ответный навык с диалоговым менеджером на правилах и fastText классификатором. Эта версия заняла второе место со средней оценкой 2.31 из 5. Победивший бот набрал 2.38, а человек 3.8. Выложенные данные содержали в себе информацию, анализ которой помог в составлении следующих шагов по развитию агента. Было принято решение добавить разговорный навык Alice, навык реферирования и определения темы. В навык приветствия было добавлено упоминание темы текста. Новые навыки требовали добавления новых правил для их вызова. Вместо правил был добавлен классификатор k-NN для навыков без диалоговых данных. Таким образом, значительно упростился процесс разработки диалоговой системы. Все это в результате привело к победе bot#1337 в финале соревнования (табл. 1).

Таблица 1 — Результаты финала ConvAI.

Бот	Качество	# Количество диалогов
bot#1337	2.779	68
poetwannabe	2.623	53
RLLChatBot	2.462	13
kAib	2.2	35
PolyU	1.75	28
DeepTalkHawk	1.262	42

Для лучшего понимания проблем бота и для разработки дальнейших шагов по улучшению, была проанализирована частота использования навыков, качество диалогового менеджера и диалоги финала. 68 диалогов системы bot#1337

были предоставлены организаторами соревнования. Полученные диалоги были вручную размечены так, что каждой реплике был проставлен навык, выданный системой (предсказанный класс), и навык, который мог бы лучше подойти под контекст диалога (истинный класс).

В табл. 2 показано распределения использования навыков bot#1337 в финале ConvAI. Они сгруппированы по оценке диалога человеком. Было посчитано сколько раз каждый тип навыка был использован в диалогах. Разговорные навыки включают в себя OpenSubtitles, Facebook и Alice. Q&A включает в себя навык ответа на вопрос, проверки ответа и генерации вопроса. Анализ табл. 2 показал, что разговорные навыки самые частотные. Соответственно, их нужно разделять на более специфичные навыки и внимательно реализовывать. Навыки реферирования и определения темы используются редко, т.к. они вызываются только один раз за весь диалог. Это не значит, что они бесполезны. Их вызов вовлекает пользователя в беседу.

Таблица 2 — Распределение вызова навыков для каждой оценки в финале ConvAI.

Оценка	Разг. нав., %	Q&A , %	Рефер., %	Опр. темы , %	Приветс., %	Общ. фразы
5	71.4	16.3	0.7	3.2	3.9	4.6
4	56.2	23.3	2.7	6.8	8.2	2.7
3	57	21	1.1	5.9	7.5	7.5
2	55.3	24.9	2.3	6.8	7.6	3
1	64.3	15.6	1.7	4.5	6.7	7.2

Имея навык, выданный системой (предсказанный класс), и вручную предоставленный навык, который мог бы лучше подойти под контекст диалога (истинный класс), была посчитана F1-мера (табл. 3).

F1 диалогового менеджера равен 0.76. Навыки генерации вопроса и реферирования, выбор которых контролировался классификатором k-NN, показывают F1 меньше навыков, выбранных классификатором fastText (разговорный навык и навык проверки ответа). F1-мера этих навыков может быть улучшена путем увеличения обучающей выборки для k-NN классификатора. Несмотря на большую выборку у навыка ответа на вопрос, его оценка качества наименьшая в табл. 3. Анализ ошибок показал, что пользователь часто задает сложный вопрос, который не покрывается обучающей выборкой.

Даже имея высокую F1-меру, требуется анализ работы каждого навыка в диалоге. Чтение диалогов необходимо для поиска ошибок в логике работы навыков и понимания в каком направлении двигаться для улучшения диалоговой системы.

После чтения 68 диалогов финала, можно сделать вывод, что наличие разговорных навыков, которые выдают разнообразные и качественные реплики, и

Таблица 3 — Качество диалогового менеджера.

Навык	Точность	Полнота	F1	Число реплик
OpenSubtitles	0.79	0.84	0.81	249
Проверка ответа	0.82	0.70	0.75	57
Alice	0.81	0.68	0.74	152
Facebook	0.63	0.85	0.73	53
Генерация вопроса	1.00	0.56	0.71	9
Реферирование	0.67	0.57	0.62	14
Ответ на вопрос	0.56	0.60	0.58	25
<i>Среднее</i>	<i>0.77</i>	<i>0.76</i>	<i>0.76</i>	<i>559</i>

вовлечение пользователя в беседу, когда он не отвечает в течение некоторого времени помогает делать диалог лучше.

Ниже представлены пункты, которые ухудшают диалог:

- Бот начинает разговор очень быстро и пользователю это не нравится.
- Пользователь спрашивает о соревновании, но у бота отсутствует навык для ответа.
- Отсутствие семантической модели не позволяет боту говорить о фактах в тексте.
- Бот выдает грубые реплики.
- Необходимо улучшать качество навыка ответа на вопрос и навыка проверки ответа.
- Бот ведет беседу в одной стилистике. Ведение диалога с использованием разных стилистик может помочь вовлечь в беседу пользователя, который повторно общается с данной диалоговой системой.

В **разделе 2.4** приведена информация о доступности разработанной системы.

Демонстрация диалогового агента доступна в Telegram по ссылке: <https://t.me/ConvAI1337Bot>. Также доступен публичный JSON API, документация к которому находится по адресу: <https://github.com/sld/convai-bot-1337/wiki/Api-Documentation>. Исходный код выложен под свободной лицензией GNU GPLv3 и доступен на GitHub: <https://github.com/sld/convai-bot-1337>.

В **разделе 2.5** приведена информация об участии bot#1337 в соревновании ConvAI 2, где вместо текста боту давалась определенная персона и бот должен был говорить о себе в ее контексте.

На следующий год после первого ConvAI, был проведен ConvAI 2 [9], который состоял из 2 раундов. Первым был раунд квалификации - недельный хакатон, по результатам которого команды проходили в финальный раунд. В отличие от ConvAI 1, здесь боту требовалось вести беседу, опираясь на выданное описание его персоны. Персона в каждом диалоге выдавалась случайно. Пример персоны: “I am a vegetarian. I like swimming. My father used to work for Ford. My favorite band is Maroon5. I got a new job last month, which is about advertising design.”. Оценка качества в раунде хакатона считалась по двум метрикам:

1. Качество диалога, как в ConvAI 1 - в конце диалога пользователь ставил боту оценку за проведенный диалог от 1 до 5.
2. Качество имитации персоны ботом - в конце диалога пользователю предлагалось выбрать один из четырех вариантов описания персоны, которую имитировал бот.

Обе оценки были нормализованы в отрезке $[0..1]$.

Для участия bot#1337 в раунде хакатона, в нем были проделаны следующие изменения:

- добавлена система проверки правописания, которая использовала библиотеку Jampell¹¹ предобученную для английского языка. Реплики пользователя могут содержать орфографические ошибки, а такая система автоматически корректирует эти ошибки.
- Были убраны навыки реферирования и определения темы, т.к. здесь нет текста о котором нужно вести беседу.
- Был добавлен навык прощания и навык выдачи информации о соревновании.
- Был добавлен навык разговора о выданной персоне. Использовалась готовая модель типа последовательность-последовательность¹², обученная на датасете Persona Chat [22], которая принимает на вход описание персоны, контекст диалога и на выходе генерирует реплику. Контекст диалога x конкатенировался с описанием персоны P : $x = \forall p \in P \oplus x$ и кодировался с помощью $h_t^e = LSTM_{enc}(x_t \mid h_{t-1}^e)$, где h_t^e - скрытое состояние энкодера в шаге t . Для кодирования слов использовались векторные представления GloVe [23]. Скрытое состояние последнего шага h_t^e подается на вход декодера $LSTM_{dec}$ в качестве его начального скрытого состояния h_0^d . Затем для каждого шага t декодер оценивает вероятность слова j через функцию softmax:

$$p(y_{t,j} = 1 \mid y_{t-1}, \dots, y_1) = \frac{\exp(w_j h_t^d)}{\sum_{j'=1}^K \exp(w_{j'} h_t^d)}.$$

- Для выбора навыка прощания и выдачи информации о соревновании был дообучен k-NN классификатор в диалоговом менеджере. Для первого навыка были добавлены такие реплики как “See you later”, “Goodbye” и т.п., для второго “What should I do in this competition?”, “What are my instructions for today?” и т.п.
- Навык разговора о выданной персоне был добавлен в fastText классификатор, т.к. была доступна исходная выборка. Также было добавлено несколько фраз, таких как “Could you tell me something interesting about you?”, “What is your dream?” и т.п. в k-NN, чтобы гарантировать срабатывание этого навыка для вопросов о персоне.

¹¹<https://github.com/bakwc/JamSpell>

¹²Пример использования можно найти здесь: <https://github.com/facebookresearch/ParlAI/tree/master/projects/convai2>

В результате бот занял 3 место в раунде хакатона (табл. 4). Это показывает обобщаемость предложенной архитектуры диалогового агента под разные задачи и контексты.

Таблица 4 — Результаты хакатон-раунда ConvAI 2.

Бот	Кач-во диалога	Кач-во имитации персоны	# Число диалогов
NEUROBOTICS	0.572	0.890	175
tensorborne	0.555	0.790	197
bot#1337	0.493	0.796	198
Lost in conversation	0.385	0.768	174
loopAI	0.479	0.610	167
baseline	0.04	0.55	81
Sonic	0.01	0.457	83

В разделе 2.6 описан механизм переходов между навыками в диалоговом агенте DREAM для Alexa Prize 2019.

Описанная выше диалоговая система использует навыки, которые не зависят друг от друга. Это обеспечивает слабую связанность системы и гибкость в разработке. Бесконтрольная реализация зависимостей между навыками может повлечь за собой многократный рост сложности системы. В данном разделе описан механизм перехода между навыками, который позволяет гибко задавать зависимость между навыками и позволяет увеличить вовлеченность пользователя в диалог. Рост сложности системы контролируется тем, что этот механизм инкапсулирован в отдельном модуле и разработчикам навыков предоставляется единый интерфейс его использования.

В 2019 году в третий раз прошло соревнование Alexa Prize Challenge. С момента окончания соревнования ConvAI 2 сильно развилась библиотека DeepPavlov и появилась новая библиотека DeepPavlov Agent для реализации диалоговых систем.

Вместо улучшения предыдущей диалоговой системы bot#1337, было принято решение разрабатывать нового диалогового агента под названием DREAM на базе DeepPavlov Agent для соревнования Alexa Prize. Использование такой библиотеки позволило ускорить разработку, проще реализовывать и поддерживать сложную логику, увеличить сопровождаемость и надежность системы, легче вводить в работу новых разработчиков. Кроме всего прочего, было предложено и реализовано множество улучшений в саму библиотеку.

Изначально связей между навыками в диалоговом агенте DREAM не было. Каждый навык вызывался в зависимости от запроса пользователя. Например, если пользователь начинает разговор о фильме, то включается соответствующий навык разговора о фильмах. Чтобы включился навык разговора о книгах, нужно было соответствующее действие от пользователя. Такое поведение ограничивает интерактивность диалога, т.к. не каждый пользователь лидирует диалог. В

какой-то момент бот должен уметь предложить переключиться на другой навык. Реализация поведения диалогового агента, когда один навык явно перенаправляет пользователя на другой, позволила значительно улучшить оценку качества диалогового агента.

Механизм перехода от *текущего* навыка к *целевому* был реализован через специальную функцию *link to*, которая работает следующим образом (рис. 5):

1. Задается словарь SKILL PHRASES, где ключом является название целевого навыка, значением *список фраз link to*, по которым может быть произведен переход на этот навык.
2. Функция *link to* принимает на вход название целевого навыка и возвращает случайную фразу из *списка фраз link to* словаря SKILL PHRASES с ключом названия навыка. Разработчики отдельных навыков используют эту функцию для перехода к целевому навыку, добавляя ее в конец сценария или в другое подходящее по логике работы навыка место.
3. Вызывается обработчик *списка фраз link to* в модуле выбора навыков (skill selector), который принимает на вход *link to фразу* бота из предыдущего шага и текущую реплику пользователя. Если обработчик на основе этой информации выбирает целевой навык, то происходит переход на следующий шаг.
4. Вызывается обработчик *списка фраз link to* в конкретном навыке. На вход обработчику идет *link to фраза* бота из предыдущего шага и текущая реплика пользователя. Обработчик возвращает конечный ответ пользователю.

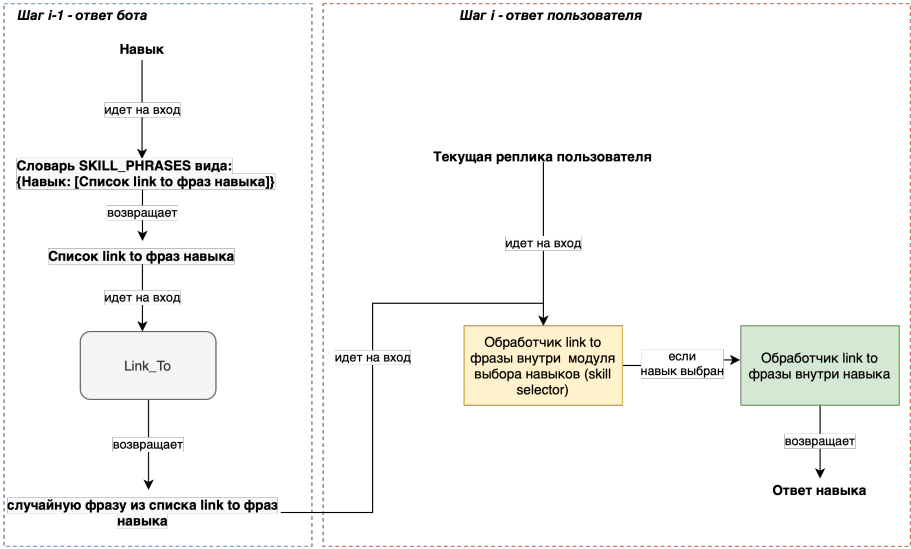


Рисунок 5 — Схема работы механизма перехода между навыками link to.

В табл. 5 приведены результаты использования механизма перехода между навыками *link to* для версии бота 9.6.0. По таблице видно, что диалоги в которых *link to* не использовался имеют наименьший рейтинг. Для диалогов, где *link to* вызвался и выдал реплику, но пользователь не проследовал в другой навык (*link to* с длиной 1), рейтинг выше, а максимальный рейтинг достигается, когда пользователь успешно продвинулся дальше используя этот механизм.

Таблица 5 — Результаты использования *link to*.

Диалоги	Рейтинг	# Количество диалогов
Без <i>link to</i>	3.29	144
С <i>link to</i> с длиной 1	3.63	124
С <i>link to</i> с длиной > 1	3.82	64

Третья глава посвящена описанию метода построения диалогового графа из диалогового корпуса и его использованию в задачах обучения с учителем.

В разделе 3.1 приведен обзор методов извлечения диалоговых графов из диалоговых корпусов.

Человек ведет диалог держа в уме определенную цель и, возможно, план по ее достижению. Например, он может беседовать с другим человеком для поддержки дружеских отношений, и в его план могут входить следующие действия: узнать как прошел день, сделать комплимент, рассказать историю и т.п. Основной задачей диалоговой системы является генерация ответа в диалоге. Есть предположение, что оценка качества в этой задаче может быть значительно улучшена, если бот будет преследовать в диалоге цель и иметь план по ее достижению.

Для получения устойчивой стратегии ведения диалога, бот должен иметь план. Такой план может быть представлен в виде *графа диалога*. Диалоговый граф показывает основные темы беседы для каждого шага и переходы между ними. Вершина графа - это основная тема беседы для данного шага в диалоге. Ребро отвечает за возможные переходы между темами. Диалоговый граф представляет из себя интерпретируемую визуализацию всего корпуса. Таким образом, граф может помочь проанализировать, почему та или иная реплика была сгенерирована диалоговой системой, использующей нейронную сеть типа последовательность-последовательность. В дополнении, диалоговый граф может быть использован как источник признаков для улучшения качества на связанных задачах.

В разделе 3.2 описывается предложенный в работе метод генерации диалогового графа, который использует комбинации различных алгоритмов кластеризации и набор эвристик. Также описана архитектура типа последовательность-последовательность и ранжирующая модель с механизмом извлечения признаков из графа.

Как определить вершину в диалоговом графе? Первоначальным вариантом может быть кластеризация по всем репликам диалогового корпуса и использование кластера как вершины графа. Однако семантика кластера в диалоге зависит от его контекста, а использование “мешка реплик” не учитывает его. Для решения этой проблемы, предлагается объединить все i -е шаги диалогов в группы, и затем проводить кластеризацию по этим группам. Таким образом, семантика каждого кластера будет привязываться к определенному шагу в диалоге.

Алгоритм построения диалогового графа представлен на рис. 6. Во первых, происходит группировка i -х шагов по всем диалогам ($i \in 1 \dots L$), где L — количество реплик в самом длинном диалоге корпуса, и их преобразование в векторное пространство. Затем производится независимая кластеризация каждой получившейся группы. После этого каждому кластеру присваивается название, а каждой реплике диалога — кластер. На последнем шаге алгоритма составляются тройки вида: название кластера предыдущей реплики в диалоге, название кластера текущей реплики, количество переходов из кластера предыдущей реплики в кластер текущей. Используя такие тройки, конструируется граф диалога.

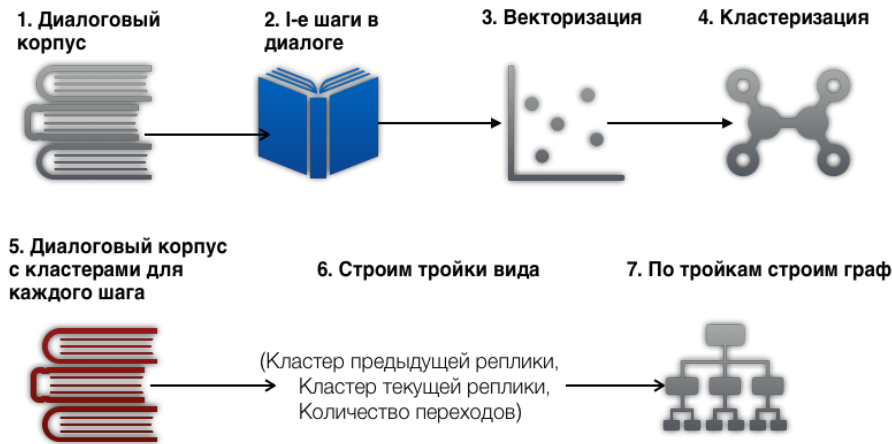


Рисунок 6 — Алгоритм построения диалогового графа.

В следующих параграфах дается детальное описание шагов построения диалогового графа.

Объединение реплик диалогов в группы На вход подается диалоговый корпус и происходит итерация по всем репликам в диалогах и их объединение в группы по шагам. Пример представлен в табл. 6.

Отображение в векторное пространство и кластеризация В данной работе описано использование tf-idf в комбинации с тематическим моделированием.

Таблица 6 — Пример объединения в группы по шагам. Имя кластера назначается как случайная реплика из группы.

Шаг	Диал. 1	Диал. 2	Группа	Назв.
1	Hi!	Hi!	[Hi!; Hi!]	Hi
2	Hello!	Hi!	[Hello!; Hi!]	Hi
3	How are you?	How are you doing?	[How are you?; How are you doing?]	How are you
4	I'm fine, thanks.	I'm good. What about you?	[I'm fine, thanks; I'm good. What about you?]	I'm fine, thanks
5	-	I'm good too, thanks.	[I'm good too, thanks.]	I'm good too, thanks
6	-	You're welcome.	[You're welcome.]	You're welcome

Также были опробованы методы кластеризации, не требующие заранее задавать количество кластеров, такие как DBSCAN, OPTICS, x-means, и трансформер BERT в качестве энкодера предложений. На них кластера явно не выделялись, либо кластера были шумными. Использование универсального энкодера предложений в комбинации с методом k-средних позволило явно выделить кластера. В качестве начального приближения имени кластера была использована случайная реплика из кластера. Такой подход ограничивал информативность названия кластера, поэтому приходилось просматривать несколько реплик и придумывать название кластера. Этот подход требует много ручной работы и поэтому было принято решение от него отказаться.

Именованние кластеров Назначение имени кластера является необходимым шагом для визуализации. Для метода k-средних можно использовать случайную реплику из кластера или ближайшую к его центроиду. При использовании тематического моделирования появляется возможность брать тему — список наиболее вероятных топ-k слов. Недостатком тематического моделирования и метода k-средних является необходимость предварительного выбора количества кластеров или тем k . Корпуса, у которых набор доменов заранее известен (например для MultiWOZ 2 - это 7), его можно использовать как k . Всего в графе $(k * L)$ вершин. Для корпусов, где набор доменов неизвестен, k можно установить эмпирически, с целью получить читаемую визуализацию.

Объединение кластеров Кластера для разных шагов могут быть похожими. Можно применять эвристику для объединения таких кластеров, когда используется тематическое моделирование: $|c_i \setminus c_j| \leq 1$, где c_i — список наиболее

вероятных топ-k слов кластера i . В случае использования k-средних, объединение кластеров не производится из-за механизма именования, описанного выше. Затем каждая реплика в диалоге аннотируется названием соответствующего кластера.

Извлечение троек После аннотации всех реплик происходит итерация по каждому диалогу и извлекаются тройки вида: <название предыдущего кластера; название текущего кластера; количество таких переходов в корпусе>. Имея такие тройки, строится матрица смежности и рисуется граф.

На рис. 7 представлен граф, построенный по примерам диалогов из табл. 6.

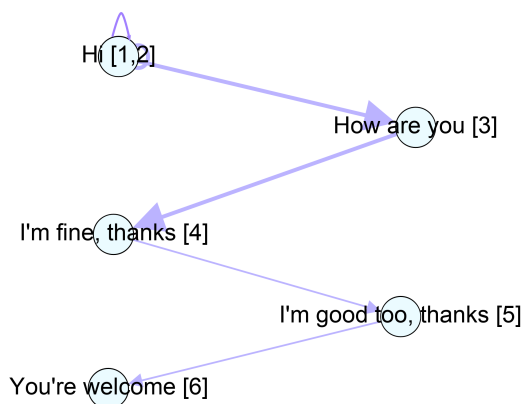


Рисунок 7 — Пример графа. Число в скобках — это шаг диалога. Толщина ребра определяется количеством таких переходов в диалогах.

Для визуализации графа была использована программа Gephi [24].

Для тематического моделирования была использована библиотека BigARTM [20].

Для кластеризации векторных представлений, полученных с помощью универсального энкодера предложений, была использована открытая библиотека Pyclustering [25]. Она включает в себя такие методы кластеризации, как DBSCAN, OPTICS, x-means, метод k-средних и т.п.

Исходный код предложенного в статье метода извлечения диалогового графа выложен под открытой лицензией MIT и доступен на GitHub¹³.

¹³<https://github.com/sld/dialog-graph-generation-experiments>

Извлечение признаков из диалогового графа Диалоговый граф также может быть источником полезных признаков, которые будут улучшать качество на целевой задаче. Для использования диалоговых графов на целевой задаче, необходимо произвести извлечение признаков, отобразив граф в фиксированное векторное представление. Предполагая, что диалоговый граф по корпусу дан, векторное представление g_i для i -го шага диалога будет строиться следующим образом:

1. Поставить в соответствие i -й реплике ее кластер и соответствующую вершину графа.
2. Преобразовать вершину графа в унитарное векторное представление (one-hot encoding) с длиной равной количеству вершин N_g , где позиция единицы считается как номер соответствующей вершины графа, а остальные номера вершин будут нулями.

Таким образом унитарное представление описывает определенную вершину в графе.

Архитектура вида последовательность-последовательность с признаками диалогового графа Признаки, извлеченные из диалогового графа, были протестированы в задаче генерации реплик диалоговой системой. Задача была поставлена как отображение контекста диалога в текущую реплику и была решена с использованием архитектуры вида последовательность-последовательность [26] с мультипликативным механизмом внимания Луонга [27]. Данные для оценки качества задачи были взяты из соответствующей дорожки по генерации реплик в MultiWOZ 2 [28]. Оттуда же была взята базовая модель, которая в декодере использует признаки из базы данных возможных диалоговых элементов и представление знаний о мире (belief state). Эти данные идут вместе с диалоговым корпусом MultiWOZ 2 и позволяют воспроизвести оценку качества базовой модели из статьи [28]. Далее, предложенная архитектура была улучшена, путем замены случайно инициализированных векторных представлений слов, которые поступают в энкодер и декодер на предобученные GLoVe [23].

Признаки, извлеченных из диалогового графа с помощью метода описанного в разделе ?? были добавлены в начальное состояние декодера по аналогии с другими признаками.

Ранжирующая модель с признаками диалогового графа Также признаки, извлеченные из диалогового графа, были протестированы в задаче выбора реплики диалоговой системой. В отличие от модели для генерации реплик, эта модель оценивает каждую реплику из списка кандидатов вместе с контекстом диалога. Реплика с максимальной оценкой используется как ответ системы. Данные для оценки качества задачи были взяты из задачи 5 корпуса bAbI 1-6.

Векторные представления можно обучать напрямую для подсчета уверенности между контекстом диалога и реплики. Обычно векторные представления обучают на текстах без учителя [29]. Такие модели учатся предсказывать слово

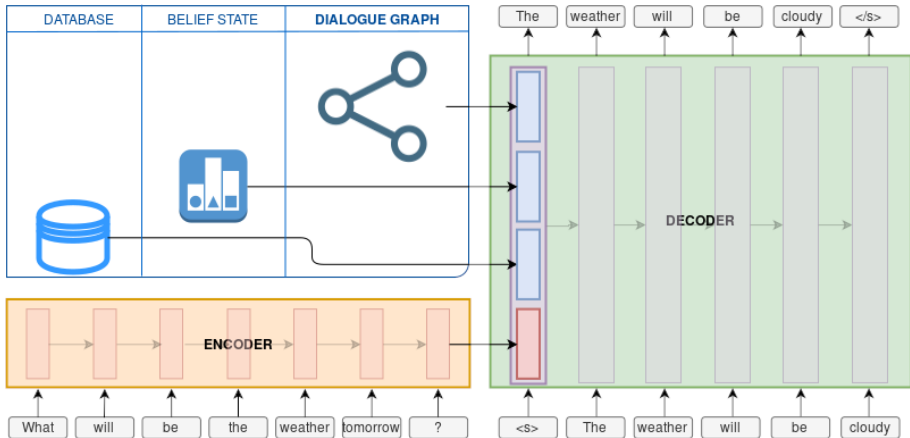


Рисунок 8 — Архитектура типа последовательность-последовательность вместе с признаками из базы данных, состояния о мире и диалогового графа. В стандартную модель типа энкодер-декодер с механизмом внимания добавлены признаки из внешних источников. Признаки конкатенированы с начальным скрытым состоянием декодера.

по его контексту в предложении и наоборот. Однако, имея обучающую выборку с диалогами можно обучать такие представления более эффективно, т.е. предсказывать следующую реплику по предыдущему контексту. Имея реплику-кандидат y , контекст x и графовые признаки g считается функция уверенности f :

$$f(x, y, g) = ((Ax)^T \oplus (Cg)^T) D) By,$$

$$A, B \in d \times V, C \in d_g \times V_g, D \in d + d_g \times d,$$

где A, B, C - матрицы весов векторных представлений слов и графовых признаков. Векторные представления обучаются через следующую функцию потерь:

$$f(x, y) \geq m + f(x, \bar{y}),$$

где m - коэффициент смещения, $\bar{y}$ - негативные кандидаты реплик. Обучение производится методом стохастического градиентного спуска [30].

В **разделе 3.3** описаны и обсуждены полученные диалоговые графы. Также представлен результат использования признаков диалогового графа на целевых задачах.

Метод был опробован на четырех различных корпусах - ConvAI 1 bot#1337, ConvAI 2, bAbI 1-6, MultiWOZ 2. Графы построенные по этим корпусам позволяют уловить основную нить беседы, хотя могут возникнуть трудности с интерпретацией имени вершины, если нет никакой экспертизы по исследуемому диалоговому корпусу. В мультидоменном корпусе MultiWOZ 2 отображены

несколько путей диалога для таких доменов как: поезд, такси, отель, ресторан и достопримечательность. Но в графе явно отсутствуют домены госпиталя и полиции. Вероятно, причиной этого является несбалансированность этого корпуса. С другой стороны в корпусах со свободным доменом ConvAI 1 и 2 присутствуют шумные вершины, значения которых невозможно однозначно интерпретировать. В корпусе с целеориентированными диалогами по одному домену bAbI 1-6 отчетливо выделился начальный этап диалога с приветствием, дальнейшие этапы выделились смазано и их трудно однозначно интерпретировать не имея знаний о корпусе.

Таким образом главная проблема построенных графов с точки зрения визуализации - это именование вершин. Один из критериев оценки тематических моделей - это перплексия. Можно отдавать названия тем (вместе с контекстами диалогов) из моделей с низкой перплексией на ручную правку, как делают авторы Graph2Bots [31], чтобы получить более качественные названия вершин.

Результаты проведенных экспериментов показывают, что диалоговый граф является источником полезных признаков, которые улучшают качество на целевой задаче. В качестве подтверждения концепции, графовые признаки были интегрированы в модель типа последовательность-последовательность и улучшено качество в задаче генерации реплики по диалоговому контексту на корпусе MultiWOZ 2. Кроме MultiWOZ2, графовые признаки были опробованы на датасете bAbI, где также выросла целевая метрика. Нельзя сказать что есть зависимость между улучшением качества и визуализацией, но в дальнейшей работе следует рассмотреть возможность одновременного решения обеих задач в рамках мультизадачной парадигмы обучения (multi-task learning) для поиска такой зависимости.

В **заключении** приведены основные результаты работы, которые заключаются в следующем:

1. Разработан диалоговый агент ведущий разговор для набора заданных тематик. Показана способность агента поддерживать, как контекст в смысле обсуждения определенного текста, так и контекст в смысле заданной персоналии агента.
2. Разработан адаптированный под холодный старт диалоговый менеджер. Для его работы требуются только данные, использованные при обучении моделей для навыков, либо несколько типовых фраз для каждого навыка.
3. Разработан комплекс программ и проведена практическая апробация в рамках международных академических соревнований. Апробация показала высокий уровень предложенного диалогового агента и диалогового менеджера по сравнению с альтернативными решениями.
4. Проанализированы диалоги, которые провели пользователи с разработанным диалоговым агентом. Анализ позволяет сделать выводы, что наличие разговорных навыков, которые выдают разнообразные и качественные реплики, а также проактивное вовлечение пользователя в

беседу, когда он не отвечает в течение некоторого времени помогает делать диалог лучше.

5. Предложен метод построения диалогового графа из диалогового корпуса для визуализации. Предложенный метод применим для корпусов разных размеров и доменных областей. Кроме визуализации диалоговый граф является источником полезных признаков, улучшающих качество решения целевой задачи.

Публикации автора по теме диссертации

- A1. Свидетельство о государственной регистрации программы для ЭВМ №2019617575 “AutoFAQ - омниканальная платформа для роботизации поддержки клиентов и сотрудников с помощью вопросно-ответных и диалоговых систем” [Текст] / И. Ф. Юсупов [и др.] // Федеральная служба по интеллектуальной собственности. — 2019.
- A2. *Yusupov, I.* NIPS conversational intelligence challenge 2017 winner system: Skill-based conversational agent with supervised dialog manager [Text] / I. Yusupov, Y. Kuratov // Proceedings of the 27th International Conference on Computational Linguistics. — 2018. — P. 3681—3692.
- A3. DREAM technical report for the Alexa Prize 2019 [Text] / Y. Kuratov, I. Yusupov, [et al.] // Alexa Prize Proceedings. — 2020.
- A4. *Юсупов, И.* Построение и использование диалогового графа для улучшения оценки качества в целенаправленном диалоге [Текст] / И. Юсупов, М. Трофимова, М. Бурцев // Труды МФТИ. — 2020. — Т. 1, № 5. — С. 75—86.

Список литературы

1. *Gao, J.* Neural approaches to conversational AI [Текст] / J. Gao, M. Galley, L. Li // The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval. — 2018. — С. 1371—1374.
2. *Traum, D. R.* The information state approach to dialogue management [Текст] / D. R. Traum, S. Larsson // Current and new directions in discourse and dialogue. — Springer, 2003. — С. 325—353.
3. *Митренина, О. В.* Прикладная и компьютерная лингвистика [Текст] / О. В. Митренина, И. С. Николаев, Т. М. Ландо. — 2016.
4. BLEU: a Method for Automatic Evaluation of Machine Translation [Текст] / K. Papineni [и др.] //. — 2002. — С. 311—318.
5. On the Evaluation of Dialogue Systems with Next Utterance Classification [Текст] / R. Lowe [и др.] // CoRR. — 2016. — Т. abs/1605.05414. — arXiv: [1605.05414](https://arxiv.org/abs/1605.05414). — URL: <http://arxiv.org/abs/1605.05414>.

6. Towards an Automatic Turing Test: Learning to Evaluate Dialogue Responses [Текст] / R. Lowe [и др.] // Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). — Vancouver, Canada, 07.2017. — С. 1116—1126. — URL: <http://aclweb.org/anthology/P17-1103>.
7. Conversational AI: The Science Behind the Alexa Prize [Текст] / A. Ram [и др.] // arXiv preprint arXiv:1801.03604. — 2018.
8. ConvAI Dataset of Topic-Oriented Human-to-Chatbot Dialogues [Текст] / V. Logacheva [и др.] // The NIPS'17 Competition: Building Intelligent Systems. — Springer, 2018. — С. 47—57.
9. The Second Conversational Intelligence Challenge (ConvAI2) [Текст] / E. Dinan [и др.] // arXiv preprint arXiv:1902.00098. — 2019.
10. Wallace, R. S. The anatomy of ALICE [Текст] / R. S. Wallace // Parsing the Turing Test. — Springer, 2009. — С. 181—210.
11. A deep reinforcement learning chatbot [Текст] / I. V. Serban [и др.] // arXiv preprint arXiv:1709.02349. — 2017.
12. The First Conversational Intelligence Challenge [Текст] / M. Burtsev [и др.] // The NIPS'17 Competition: Building Intelligent Systems. — Springer, 2018. — С. 25—46.
13. Opennmt: Open-source toolkit for neural machine translation [Текст] / G. Klein [и др.] // arXiv preprint arXiv:1701.02810. — 2017.
14. Kingma, D. P. Adam: A method for stochastic optimization [Текст] / D. P. Kingma, J. Ba // arXiv preprint arXiv:1412.6980. — 2014.
15. Tiedemann, J. News from OPUS-A collection of multilingual parallel corpora with tools and interfaces [Текст] / J. Tiedemann // Recent advances in natural language processing. T. 5. — 2009. — С. 237—248.
16. Neural Question Generation from Text: A Preliminary Study [Текст] / Q. Zhou [и др.] // National CCF Conference on Natural Language Processing and Chinese Computing. — Springer. 2017. — С. 662—671.
17. SQuAD: 100,000+ questions for machine comprehension of text [Текст] / P. Rajpurkar [и др.] // arXiv preprint arXiv:1606.05250. — 2016.
18. The Stanford CoreNLP natural language processing toolkit [Текст] / C. Manning [и др.] // Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations. — 2014. — С. 55—60.
19. Bidirectional attention flow for machine comprehension [Текст] / M. Seo [и др.] // arXiv preprint arXiv:1611.01603. — 2016.
20. Bigartm: Open source library for regularized multimodal topic modeling of large collections [Текст] / K. Vorontsov [и др.] // International Conference on Analysis of Images, Social Networks and Texts. — Springer. 2015. — С. 370—381.

21. Bag of tricks for efficient text classification [Текст] / A. Joulin [и др.] // arXiv preprint arXiv:1607.01759. — 2016.
22. Personalizing Dialogue Agents: I have a dog, do you have pets too? [Текст] / S. Zhang [и др.] // Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). — 2018. — С. 2204—2213.
23. *Pennington, J.* GloVe: Global Vectors for Word Representation [Текст] / J. Pennington, R. Socher, C. D. Manning // Empirical Methods in Natural Language Processing (EMNLP). — 2014. — С. 1532—1543. — URL: <http://www.aclweb.org/anthology/D14-1162>.
24. *Bastian, M.* Gephi: an open source software for exploring and manipulating networks [Текст] / M. Bastian, S. Heymann, M. Jacomy // Third international AAAI conference on weblogs and social media. — 2009.
25. *Novikov, A.* PyClustering: Data Mining Library [Текст] / A. Novikov // Journal of Open Source Software. — 2019. — Апп. — Т. 4, № 36. — С. 1230. — URL: <https://doi.org/10.21105/joss.01230>.
26. *Sutskever, I.* Sequence to Sequence Learning with Neural Networks [Текст] / I. Sutskever, O. Vinyals, Q. V. Le // CoRR. — 2014. — Т. abs/1409.3215. — arXiv: [1409.3215](https://arxiv.org/abs/1409.3215). — URL: <http://arxiv.org/abs/1409.3215>.
27. *Luong, M.-T.* Effective Approaches to Attention-based Neural Machine Translation [Текст] / M.-T. Luong, H. Pham, C. D. Manning // CoRR. — 2015. — Т. abs/1508.04025. — arXiv: [1508.04025](https://arxiv.org/abs/1508.04025). — URL: <http://arxiv.org/abs/1508.04025>.
28. MultiWOZ - A Large-Scale Multi-Domain Wizard-of-Oz Dataset for Task-Oriented Dialogue Modelling [Текст] / P. Budzianowski [и др.] // CoRR. — 2018. — Т. abs/1810.00278. — arXiv: [1810.00278](https://arxiv.org/abs/1810.00278). — URL: <http://arxiv.org/abs/1810.00278>.
29. Distributed representations of words and phrases and their compositionality [Текст] / T. Mikolov [и др.] // Advances in neural information processing systems. — 2013. — С. 3111—3119.
30. *Bordes, A.* Learning end-to-end goal-oriented dialog [Текст] / A. Bordes, Y.-L. Boureau, J. Weston // arXiv preprint arXiv:1605.07683. — 2016.
31. Graph2Bots, Unsupervised Assistance for Designing Chatbots [Текст] / J.-L. Bouraoui [и др.] // Proceedings of the 20th Annual SIGdial Meeting on Discourse and Dialogue. — 2019. — С. 114—117.

Юсупов Идрис Фаргатович

Контекстный диалоговый агент

Автореф. дис. на соискание ученой степени канд. техн. наук

Подписано в печать _____._____._____. Заказ № _____

Формат 60×90/16. Усл. печ. л. 1. Тираж 100 экз.

Типография _____

