МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

**«РОССИЙСКИЙ ГОСУДАРСТВЕННЫЙ ГУМАНИТАРНЫЙ УНИВЕРСИТЕТ»
(РГГУ)**

ИНСТИТУТ ЛИНГВИСТИКИ
Кафедра (учебно-научный центр) Компьютерной лингвистики.....

Генералова Валерия Антоновна

Автоматизированное извлечение реплик, свидетельствующих об отрицательной оценке собеседника, из человеко-машинных диалогов различной тематики

Выпускная квалификационная работа студентки 2-го курса

очной формы обучения

Направление 45.04.03 Фундаментальная и прикладная лингвистика

Направленность Компьютерная лингвистика

Допущена к защите на ГЭК

Заведующий кафедрой
В. П. Селегей

_____ (Ф.И.О.)

«____»_____ 20….. г.

Научный руководитель
С. А. Шаров

_____ (Ф.И.О.)

«____»_____ 20….. г.

Москва 2018

Введение 3
Раздел 1: История вопроса 9
Раздел 2: Сбор материала 15
Постановка проблемы 15
Чатбот ρBot 17
Чатбот Маришко 18
Инфы 18
Форум инфов 20
Инф Кармен 22
Инф Aplgallery 22
Инф Alisa_fpml 23
Сравнение источников 23
Первичная обработка материала 25
Раздел 3: Создание признакового пространства 27
Постановка проблемы 27
Автоматический метод: doc2vec 28
Контролируемый метод 36
Гибридный метод 44
Раздел 4: Классификация 48
Постановка проблемы 48
Обзор существующих решений 50
Создание и обучение моделей 52
Подготовка выборки 52
Выбранные методы 53
Autoencoder 53
One-Class Support Vector Machine 54
Elliptic Envelope 55
Local Outlier Factor 56
Isolation Forest 56
Сводная таблица методов 57
Применение лучших моделей ко всему корпусу 58
Раздел 5: Выводы 59
Приложение 1 61
Приложение 2 63

Введение

Системы, обеспечивающие общение человека с компьютером на естественном языке, в последние годы приобрели невероятную популярность. В русскоязычном мире событием стало появление Яндекс.Алисы – голосового помощника для мобильных и настольных устройств, способного находить файлы, решать повседневные задачи и поддерживать разговор на любую тему[1].

В связи с увеличением количества систем, обеспечивающих человеко-машинный диалог, и ростом их аудитории остро встаёт вопрос оценки диалоговых систем. Несмотря на давнюю историю вопроса (см. Раздел 1), в настоящее время не существует метода однозначной объективной оценки качества диалоговой системы. Нельзя не отметить, что затруднительность создания метода оценки качества главным образом обусловлена размытостью понятия «качество диалоговой системы».

Поскольку диалоговые системы зачастую выполняют роль интерфейса и ориентированы в первую очередь на пользователей, удовлетворённость пользователя часто принимается за мерило качества системы.

Настоящее исследование посвящено **теме** автоматизированного извлечения реплик, свидетельствующих об отрицательной оценке собеседника, из человеко-машинных диалогов различной тематики.

Цель исследования заключается в том, чтобы с помощью лингвистических и компьютерных методов правильно определить искомые реплики и точно извлечь их из коллекции имеющихся данных.

Объектом исследования являются реплики, свидетельствующие об отрицательной оценке собеседника. Под этим термином понимаются реплики (высказывания), содержащие недовольство либо собеседником в целом (см. Пример 1 в Таб. 1), либо коммуникативной ситуацией (см. Пример 2 в Таб. 1), либо отдельными параметрами собеседника (см. Пример 3 в Таб. 1)или ситуации (см. Пример 4 в Таб. 1).

[1] Доклад Бориса Янгеля, группы разработки диалоговых систем, на внутренней конференции Яндекса: https://habr.com/company/yandex/blog/349372/ (Дата обращения: 10.06.2018)

Пример 1
X: дивное словцо юля , однако , из уст твоих излетело Y: хватит ужу ты достала X: хи - хи - хи Y: ты меня задрадла кошка ты
Пример 2
X: правда что ли ? хи - хи - хи . нет . но , кроме шуток . скажи . правда ? Y: ты про что ? X: что , " что " ? что - нибудь не так ? или , нормально ? Y: я тебя не понимаю
Пример 3
X: я подумаю над этим на досуге ! Y: подумай - подумай . тебе нужно развивать мозг X: ну и почерк у тебя , пишешь , просто как бройлер окорочком ! Y: может . просто ты читать не умеешь ?
Пример 4
X: считается , что единственный ребенок в семье может вырасти эгоистом Y: ты такие сложные выражения используешь , я их не знаю X: надо будет еще с кем - нибудь это обсудить Y: с кем - нибудь

Таб. 1 Примеры диалогов, содержащих реплики, свидетельствующие о недовольстве пользователя

Следует отметить, что под недовольством в данном случае понимается широкий спектр различных негативных эмоций, от непонимания до гнева, включая прямые оскорбления и ироничные рекомендации. Для обозначения подобных реплик в англоязычных источниках используется термин *negative feedback* (подробнее о термине см. в Разделе 1), в русскоязычных встречается калька *отрицательная обратная связь*. Все эти понятия обозначают некоторые языковые явления, содержащие некоторые элементы, направленные на взаимодействие с собеседником и выражающие отрицательное отношение к чему-либо, связанному с ситуацией. Это определение ни в коей мере не претендует на формальное, однако некоторым образом очерчивает круг искомых элементов.

Материалом исследования послужили диалоги, созданные в режиме реального времени (через специальные интерфейсы на сайтах в режиме мгновенных сообщений или через сообщения в социальных сетях) в текстовом виде на русском языке.

Материал ограничен только текстовыми сообщениями и не учитывает голосовое общение между человеком и компьютером по ряду причин. Во-первых, сбор голосовых данных сложен с методологической и этической точки зрения (подробнее см. исследования, использующие методы проекта «Один речевой день» [Хониева 2016]). Во-вторых, расшифровка голосовых данных ресурсоёмка, так как требует длительного времени работы

эксперта. В силу этих двух особенностей корпуса устных человеко-машинных диалогов крайне малы по объёму. В-третьих, при голосовом общении люди, как правило, не ведут длинных диалогов, поскольку необходимый для этого алгоритм действий (нажать кнопку – записать свой голос – убрать кнопку – подождать ответа – услышать ответ) всё же ещё не совсем естествен для повседневного общения – в отличие от мгновенных текстовых сообщений, занимающих важное место в картине ежедневной коммуникации. Было замечено, что голосом люди чаще всего пользуются для решения конкретных оперативных задач (позвонить, найти дорогу, внести событие в календарь). Свободное общение с голосовыми ассистентами чаще всего строится по парам вопрос-ответ, не представляя собой целостного диалога. Все эти особенности делают голосовые диалоги между людьми и компьютерными диалоговыми системами не совсем подходящим материалом для исследования в рамках заявленной темы. Однако не исключено, что в отдалённой перспективе или с определёнными поправками будет возможно провести подобное исследование и на материале устных диалогов.

Выбор языка диалогов также нуждается в пояснении. Большинство исследований основаны на англоязычном материале, что имеет ряд преимуществ. С методологической точки зрения выбор английского языка удобен, поскольку существует немало готовых коллекций данных и уже натренированных моделей для работы с ними, что позволяет не тратить время и усилия на разработку новых методов и сосредоточиться на лингвистических наблюдениях. С компьютерно-лингвистической точки зрения английский язык удобен, поскольку его морфологическое и синтаксическое разнообразие относительно невелико, что позволяет получать более точные результаты при автоматизированной обработке. С практической лингвистической точки зрения работа с английским языком перспективна, поскольку большое количество речи в интернете создается именно на нём, и изучение специфики выражения недовольства может оказаться полезным для различных других исследований. Тем не менее, мы предполагаем, что введение в научный и практический обиход русскоязычных материалов не только оправдано, но и необходимо. Русскоязычное интернет-сообщество и русскоязычный мир в целом достаточно велики, чтобы его изучение имело не только теоретический, но и практический интерес. Это требует развития методологии и инфраструктуры (корпусов, программ и др.), учитывающих специфику русского языка, в частности, его синтаксические и морфологические особенности. Мы предполагаем, что некоторые результаты настоящего исследования смогут быть полезными для развития в этом направлении.

Отдельно следует пояснить, что задача сопоставления результатов, созданных на различном материале, в настоящем исследовании не ставилась. Во-первых, данные могут

быть собраны по различным принципам, что затруднит их сравнение. Во-вторых, могут присутствовать так называемые скрытые переменные, которые не учитываются в каждом из отдельных исследований, но могут играть роль при сопоставлении. В-третьих, мы полагаем, что без достаточно тщательного изучения материалов одного типа, не следует переходить к сравнению, поскольку достоверность результатов может вызывать сомнения.

Выбор материала также обусловлен выбором тематики и цели диалогов. Традиционно производится деление диалоговых систем на целеориентированные (goal-oriented) и не целеориентированные (non goal oriented). Первых систем в настоящее время значительно больше, чем вторых, и они постоянно создаются. Вторые системы сложнее и интереснее для анализа. В нецелеориентированных системах покрываются различные темы, которые могут меняться в процессе одного диалога. Важнейшим отличием таких систем является отсутствие естественного критерия успешности диалога. В то время как при решении конкретной задачи успешным можно считать случай, когда задача решена, и неуспешным случай, когда задача не решена, в режиме свободного диалога нельзя сказать, в какой момент произошла коммуникативная неудача (если вообще произошла). Мы полагаем, что в свободных диалогах различной тематики недовольство одного из собеседников оказывается важнейшим критерием оценки качества диалога. Именно поэтому настоящее исследование основывается на материале именно таких систем.

Актуальность темы обусловлена актуальностью материала и новизной подходов. Большинство существующих работ по теме оценки диалоговых систем написаны на материале английского, в то время как настоящее исследование выполнено на материале русского языка. Также большинство работ выполнено на материале целеориентированных (goal-oriented) диалоговых систем, в то время как настоящее исследование основывается на не целеориентированных (non goal-oriented). Существующие методы оценки диалоговых систем используют автоматизированные алгоритмические решения для учёта технических характеристик диалога (количество реплик, среднее время ожидания ответа и др., подробнее см. в [Möller 2005]), однако целью настоящего исследования является автоматизированное извлечение некоторых реплик на основе их содержания (реплик, объединённых общей тематикой неудовлетворённости).

Исследование имеет **теоретическую и практическую значимость**. В лингвистической теории результаты работы могут помочь лучше понять специфику выражения недовольства в современном русском языке и особенности употребления различных оценочных номинаций людей. Также работа способствует развитию изучения человеко-машинных диалогов как новой формы коммуникации. Эти исследования могут быть полезны как для специалистов по изучению диалогической речи, так и для

специалистов, изучающих поведение людей в различных коммуникативных ситуациях. Значимость исследования для компьютерной лингвистики состоит в том, что оно представляет подробное исследование некоторых методов классификации и предлагает поправки для их применения в нестандартных ситуациях. Впоследствии методы с этими поправками можно будет использовать для решения других задач классификации неразмеченных несбалансированных данных.

Практическая значимость работы состоит в том, что исследование является шагом к ещё не созданной, но уже очень востребованной на рынке системе автоматической оценки качества диалога. Системы автоматизированного определения реплик, свидетельствующих о неудовлетворённости, помогут компаниям, владеющим диалоговыми системами, узнавать о случаях недовольства в режиме реального времени и, возможно, применять адаптивные стратегии для совершенствования диалоговых систем. Кроме того, автоматизированный поиск неудовлетворительных коммуникативных ситуаций позволит компаниям получать более полную информацию об успешности их систем. Конечно, в настоящее время во многих системах реализована возможность эксплицитного выражения пользовательского отношения к системе: под репликами диалогового агента (бота) появляются кнопки «нравится» и «не нравится». Однако люди используют их далеко не во всех случаях, когда не удовлетворены репликой бота. Автоматизированное извлечение подобных реплик снимет нагрузку и с пользователей, и с разработчиков подобных систем отзыва.

В соответствии с поставленной целью исследование предполагает решение ряда **задач**, коррелирующих с этапами исследования:

1. Ознакомиться с литературой по теме оценки диалоговых систем и методам классификации.
2. Создать корпус материалов.
3. Представить тексты в численном виде.
4. Оценить приемлемость методов для достижения цели и выбрать лучшие.
5. Обучить компьютерные модели и оценить их результативность на небольшой выборке.
6. Выбрать лучшую модель и применить её ко всему корпусу.
7. Оценить качество извлечения реплик.

Работа организована в соответствии с хронологическим (и логическим) порядком решения поставленных задач. Раздел 1 содержит обзор литературы. Раздел 2 посвящён описанию процедуры сбора и предварительной обработки материалов. Раздел 3 представляет методы, использованные для преобразования текстов в численные векторы и

освещает некоторые промежуточные результаты, связанные с оценкой применяемых методов. Раздел 4 подробно описывает методы классификации текста, содержит обоснование выбора методов для настоящего исследования, а также результаты. Раздел 5 суммирует и интерпретирует результаты, а также предлагает ряд задач для дальнейших исследований. Работа завершается списком процитированной литературы и Приложением.

Раздел 1: История вопроса

Вопрос оценки диалоговых систем имеет достаточно давнюю историю. В настоящем разделе отмечаются некоторые тенденции в исследовании этой предментой области, а также приводятся ссылки на наиболее важные и интересные работы теоретического характера. Перечисление исчерпывающего списка работ, посвящённых теме анализа диалога, кажется не только чрезвычайно затратным (ввиду существования множества небольших модификаций систем оценки применительно к каждой отдельной диалоговой системе), но и неуместным, поскольку все конкретные системы опираются на общие теоретические принципы. Вторая часть раздела посвящена вопросу извлечения реплик, свидетельствующих о недовольстве одного из собеседников (negative feedback).

Следует отметить две важные особенности существующего положения дел. Во-первых, гораздо больше работ посвящено целеориентированным (task-oriented) системам, а не тем, которые ведут диалог на свободную тему. Во-вторых, для анализа собственно диалогов (а не метаданных) без помощи людей на том или ином этапе обойтись до сих пор не удавалось. Эти тенденции формируют актуальность настоящего исследования.

Основной пласт литературы, посвящённой оценке диалоговых систем, создан в 1995–2005 гг. Именно в это время развиваются некоторые общие положения теории оценки диалога вообще или отдельных модулей конкретных. Например, в [Paek 2001] содержится подробное рассуждение о том, зачем вообще оценивать диалоговые системы, и предлагается использовать различные методы в зависимости от цели. Основным методом оказывается так называемый Wizard of Oz, система, в которой два человека ведут диалог, считая, что говорят с машиной. Автор предлагает заменять на симулятор последовательно каждый из модулей диалоговой системы, чтобы облегчить тестирование. Всю процедуру предлагается проводить в четыре этапа: 1) выбрать метрику, пригодную для объективной оценки, 2) изменять компонент, который бы наилучшим образом проверялся выбранной метрикой, 3) оставить остальные части неизменными, 4) повторить, используя различных «волшебников». Несмотря на тщательность методологической проработки решения, оно кажется слишком затратным с точки зрения использования человеческих усилий. Однако на начальном этапе развития отрасли диалоговых систем не было ни технологий, способных значительно сократить эти усилия, ни соответствующего отношения к компьютерным методам как к чему-то, что действительно способно справляться со сложными даже для человека задачами.

В 1997 году появилась схема оценки PARADISE ([Walker et al. 1997]), которая предлагала оценивать удовлетворённость пользователя ответом. Для этого предлагалось считать 1) успешность выполнения задания, 2) эффективность диалога (чем меньше шагов

и времени, тем лучше), 3) качество ответов системы (учитывались ответы, требовавшие долгого времени, неадекватные ответы и др). Успех задания оценивается с помощью матриц ценности атрибутов (attribute value matrix, AVM), состоящими из параметров, которые должны прозвучать в диалоге, и их возможных значений. Коэффициент успеха рассчитывается с учётом количества совпадающих «правильных ответов», т.е. случаев, когда значения из матрицы совпадают с теми, которые были определены по сценарию и с поправкой на количество совпадений, которые можно было бы ожидать случайно. Размечать правильные и неправильные ответы в матрицах предлагается заранее либо людям, либо компьютерам, но с последующей проверкой человеком. Оценивать стоимость диалогов предлагается с помощью иерархии целей диалога, которая тоже создаётся людьми и превращается в матрицу. Наконец, качество диалога предлагают оценивать по опросам, предложенным пользователям после взаимодействия с диалоговой системой. Как видно, эта система оценки рассчитана на диалоги, у которых есть цель, достижимая тем или иным путём, но можно приложить её методы и к частям системы, не ориентированной на выполнение конкретных заданий. И действительно, эта работа была взята за основу ряда других исследований.

Пример использования этой системы можно найти, например, в [Litman et al. 1998]. В этой статье пользователи тестируют две версии одной и той же системы, которая затем оценивается в соответствии с критериями PARADISE. В дальнейших версиях той же системы ([Litman & Pan 1999]) применяется ещё и статистический тест ANOVA для оценки адаптивности системы, однако отправной точкой для всех оценок оказываются данные, полученные от пользователей.

На основе системы PARADISE проводилось множество исследований, были попытки её улучшить или добавить в неё новые методы. Некоторые старались найти способ измерять качество диалога без опроса пользователей, что привело к развитию систем, старающихся предсказать заранее лучший ответ. Например, была идея применить методику collaborative filtering, которая использует данные из специально собранного корпуса, чтобы предсказать ответ ([Yang et al. 2012]). В представленной в статье версии системы корпус, тем не менее, размечался пользователями. Но последующее обучение модели происходило без вмешательства человека.

На основе подобных корпусов, заранее собранных с помощью Wizard of Oz и вручную размеченных, оценивались и симулированные диалоги, т. е. диалоги, имитирующие человеческие, но происходящие между двумя частями диалоговой системы. В проекте DIHANA ([Griol et al. 2012]) предлагалась оценка, основанная исключительно на количественных показателях. Система на основе статистических закономерностей

генерировала корпус диалогов, который затем сравнивался с исходным. Собственно, именно в сравнении двух корпусов диалогов и заключается новизна метода.

Авторы предлагали оценивать 1) количество успешных, т. е. достигших цели, диалогов, 2) среднее количество реплик на диалог, 3) количество различных успешных диалогов, 4) количество реплик в самом коротком диалоге, 5) количество симулированных диалогов, содержащихся в исходном корпусе. Параметры 1 и 3 похожи с той лишь разницей, что параметр 3 оценивает способность системы создавать разнообразные диалоги, тем самым оценивая ниже переобученные системы или системы с малым количеством правил. Параметры 2 и 4 оценивают успешность диалога по его количественным характеристикам, опираясь на два противоположных допущения: если диалоги слишком длинные, значит, с системой общаться неудобно и цели достигнуть трудно, но если встречаются очень короткие диалоги, значит, бывают такие ситуации, когда система с самого начала ошибается и взаимодействие с ней прекращается. Параметр 5 специфичен именно для метода, сравнивающего два корпуса диалогов.

Метод сравнения сгенерированного корпуса с обучающим был применён для оценки обучаемости системы в рамках того же проекта. Для этого были использованы четыре меры: 1) процент ситуаций, отсутствовавших в обучающем корпусе, но встретившихся в тестовом (чем меньше, тем более надёжна система), 2) процент ответов, следующих стратегии из обучающего корпуса (набор стратегий и правил был заранее размечен, достаточно просто посчитать количество совпадений), 3) процент ответов, не следующих приписанной стратегии, но всё равно связных (считается вручную, новые данные можно добавить в обучающий корпус), 4) процент ответов, повлекших за собой прекращение диалога (тоже оценивается вручную). Сама по себе идея такого сравнения интересна, но вовлечённость человека всё же велика.

Идея сравнения двух корпусов легла в основу и других систем оценки, одна из которых представлена в статье [Eckert et al. 1998]. Для каждого диалога находится комплексная мера, основанная на весах его компонентов. Однако как именно рассчитывается эти веса, не указано. Общая мера для диалоговой системы D, взаимодействующей с пользователями U, рассчитывается на основании мер q для входящих в неё диалогов d, а затем сравнивается с корпусом C (подробнее см. стр. 4 указанной работы). Представленный метод отличается от метода PARADISE как внутренней логикой, так и набором данных, необходимых для оценки, однако не получил большой популярности.

Для оценки диалоговых систем применялись также методы из других областей компьютерной лингвистики, в частности, машинного перевода и автоматической

экстракции информации. Наиболее полный (и достаточно недавний) обзор представлен в [Liu et al. 2016].

Также попытку отступить от традиционных мер оценки связности диалога сделали авторы работы [Gandhe & Traum 2008]. Они отметили, что попытка предсказать следующую реплику в диалоге сводится к попытке предсказать следующий элемент последовательности, а эта задача успешно решается в рамках суммаризации текстов. Поэтому авторы предложили использовать меру τ Кендалла, а также биграммы и триграммы. В статье подробно описано сравнение этих мер. В целом, биграммы и триграммы оказались продуктивнее τ, хотя последняя мера и лучше показывает наличие взаимосвязи на расстоянии, что как раз могло бы оказаться полезным при анализе диалогов на свободную тему, где реплики, связанные одним контекстом, не всегда идут подряд.

Отдельно выделяется направление исследований, занимающихся типологией ошибок, возникающих в диалоговых системах и (иногда) предлагающих способы решения некоторых из них. Эти работы очень важны и, несмотря на то, что современные технологии используют другие методы, актуальны по сей день, поскольку правильное понимание ошибки может помочь разработать новые методы её исправления. В таких работах диалоговые системы рассматриваются по модулям и ошибки предлагается исправлять также отдельно в каждом модуле. Наиболее интересными для цели настоящего исследования оказываются ошибки, связанные с ведением диалога.

Промежуточным между описанными подходами можно считать исследование, описывающее систему генерации диалоговых актов, принимая во внимания данные из агентов, каждый из которых отвечает за свой уровень оформления реплики ([Keizer & Bunt 2007]). Кандидаты от каждого уровня берутся отдельно и сравниваются. Те, которые как-либо связаны друг с другом, получают преимущества, а те, которые конфликтуют, отбрасываются. Примечательно, что такая система позволяет оценивать любые реплики, как содержательные, так и служебные.

Тщательно разработана типология ошибок в соответствии с уровнями диалога в исследовании [Higashinaka et al. 2015]. Авторы выделяют ошибки в построении высказывания, в выборе высказывания, в соответствии высказывания контексту диалога и общей ситуации. Для того, чтобы оценить состоятельность этой методики, авторы попросили пользователей разметить диалоги и получили сведения о том, как ошибки разных типов связаны друг с другом. Тщательность проработки самой системы впечатляет, однако пока не ясно, как применять эту типологию ошибок.

Однако наиболее проработанной кажется система, представленная в статье [Möller 2005]. Собственно, в статье гораздо более важным оказывается приложение, а не

сама статья. В нём представлены выявленные авторами параметры взаимодействия (interaction parameters) и приводятся способы их оценки (инструментальный или экспертный). Выделяемые ошибки не зависят от архитектуры системы, что даёт этой таксономии перспективу стать основой универсальной системы оценки. Однако многие параметры (особенно связанные с ведением диалога) всё же предлагается оценивать экспертам, что было бы недопустимо в универсальной системе автоматической оценки диалоговых систем.

Отдельно стоит упомянуть несколько исследований, связанных с извлечением реплик, свидетельствующих об отрицательной оценке собеседника (negative feedback), из корпусов текстов.

Чаще всего работы по этой теме посвящены взаимодействию с обучающими системами. Под термином feedback подразумеваются отзывы на реплики учащихся. Под термином negative feedback в таких случаях понимается отзыв на ответ, эксплицитно или имплицитно указывающий на присутствие в реплике учащегося какого-либо нецелевого элемента. Отмечается, что negative feedback лишь указывает на наличие проблемы, но не предлагает способов её решения[2] ([Iwashita 2003]). С некоторой долей обобщения это определение применимо и к другим областям. Перцепция элемента как нецелевого может быть обусловлена не необходимостью дать верный ответ на вопрос, но какими-то другими (в том числе субъективными) лингвистическими и когнитивными факторами. В этом случае также исчезает разница между «учеником» и «учителем», поскольку оба собеседника могут интерпретировать ответы другого в соответствии с собственными стандартами. При этом могут возникать комплексные единицы, одновременно имплицитно указывающие на наличие проблемы и эксплицитно предлагающие выход из сложившейся ситуации (реплики вроде «Не могли бы Вы переформулировать вопрос?»).

В применении к автоматической обработке речи термин negative feedback появляется в [Brennan & Hulteen 1993]. Авторы заостряют внимание на том, что своим успехом система голосового взаимодействия обязана не только точности распознавания голоса. Они выделяют семь стадий взаимодействия человека с машиной и полагают, что после завершения наиболее релевантной из них пользователь должен получить отчёт об успешности или неуспешности действия (positive или negative feedback соответственно). Авторы подробно останавливаются на том, какая именно стадия оказывается наиболее релевантной и заслуживающей того, чтобы после неё пользователь получил отчёт.

[2] Впрочем, не все исследователи разделяют такое определение понятия negative feedback (см., в частности, [Gass 1997]).

Понимание negative feedback как отчёта о провале задания вполне соотносится с определением из области изучения второго языка, представленным выше, и может быть применено к нецелеориентированным диалогам. Действительно, в обычном диалоге пользователь тоже получает отчёт после каждой реплики. Положительным отчётом в таком случае оказывается просто следующая реплика диалога, разворачивающегося без помех. Сигналом о помехе может служить имплицитное или эксплицитное указание на несоответствие ответа ожиданиям собеседника.

До сих пор термин feedback понимался как информация, сообщаемая системой пользователю, однако нет никаких препятствий для того, чтобы оценивать информацию, сообщаемую пользователем системе по тем же параметрам. Так, в [Bell & Gustafson 2000] такая информация анализируется и оценивается по трём параметрам. Первый параметр оценивает тональность сообщения, указывая, положительный он или отрицательный. Авторы отмечают, что решающим фактором является лексический, однако в случае неоднозначности лексического элемента в оценке тональности могут помочь просодические характеристики. Второй параметр оценивает, сообщается ли информация эксплицитно или имплицитно. Приводимые в статье примеры не дают однозначно понять характеристик, по которым авторы относят высказывания к тому или иному типу, однако и сами авторы признаются, что есть немало неоднозначных случаев. Третий параметр оценивает, что именно сообщается: либо просто указание на то, что информация от системы получена (одобрена, не одобрена, нуждается в модификации), либо отношение говорящего (пользователя) к системе или к коммуникативной ситуации. Авторы отмечают, что наблюдается варьирование в выборе стратегии в зависимости от окружающего контекста и индивидуальных предпочтений собеседников.

В настоящем исследовании не производится дифференциации между эксплицитными и имплицитными сообщениями. Рассматриваются только те реплики, которые относятся к негативным по первому параметру и входят в любую группу по третьему параметру.

Попытки автоматической обработки подобной информации предпринимались ранее. В частности, в [Gamon 2004] производится попытка автоматической классификации пользовательских реплик по тональности. В статье комбинируются методы машинного обучения и лингвистического анализа для обеспечения лучшего результата. Автор отмечает, что результаты классификации оказываются значительно ниже, чем результаты по другим материалам ввиду сложности исследуемых данных. Эти наблюдения очень ценны для оценки успешности нашего исследования.

Раздел 2: Сбор материала

Постановка проблемы

Сбор материала представлял отдельную задачу. В настоящее время не существует открытых коллекций данных (*dataset*), содержащих человеко-машинные диалоги на русском языке. Как правило, русскоязычные исследования либо представляют авторские или коммерческие разработки ([Дегтева 2015]), либо выполняются на материале английского языка ([Асиновская 2016]). Не-русскоязычных работ, посвящённых русскоязычным человеко-машинным диалогам, также найти не удалось. Из этого следует необходимость создания открытого русскоязычного корпуса человеко-машинных диалогов, на котором в дальнейшем можно было бы проводить исследования, подобные настоящему. Сразу следует отметить, что материал настоящего исследования представляет собой скорее коллекцию данных, а не корпус, поскольку лишён какой бы то ни было разметки. Создание полноценного корпуса диалогической речи и отдельного подкорпуса человеко-машинных диалогов представляется отдельной важной и интересной задачей.

Конечно, самым подходящим источником были бы записи (аудиозаписи или транскрипты) разговоров различных людей с современными программами-ассистентами. Такой корпус отражал бы наиболее современные технологии и наиболее распространённые реакции пользователей на ошибки диалоговых систем. Кроме того, универсальность, по меньшей мере, постулируемая, таких систем обеспечивала бы широкий спектр тематики диалогов. Однако компании, разрабатывающие эти программы, не предоставляют доступ к подобным данным. Ведутся исследования, записывающие разговоры с мобильными интеллектуальными ассистентами по методологии «Один речевой день», однако получаемые корпуса в силу объективных причин невелики. Кроме того, разговоры, которые люди устно ведут с ассистентами, скорее представляют собой набор пар вопрос-ответ, а не длинный выстроенный диалог. Поэтому было принято решение отказаться от погони за новейшими данными в пользу обретения большего объёма более подходящих диалогов.

Некоторые ресурсы в сети предлагают использовать корпуса диалогов, полученных из художественной литературы. Несмотря на то, что использование подобных корпусов в настоящем исследовании помогло бы получить значительное количество данных, такие диалоги кажутся неприемлемыми по ряду причин. Во-первых, общение в интернете чаще всего реализуется с использованием разновидности русского языка, далёкой от литературного. Во-вторых, общение с диалоговыми системами (ботами) сопряжено с рядом технических сложностей, что отличает такое общение от человеческого. В третьих, круг тем, обсуждаемых с чат-ботами, отличается от тематики художественных произведений.

Ввиду этих особенностей было принято решение о самостоятельном сборе коллекции данных для исследования.

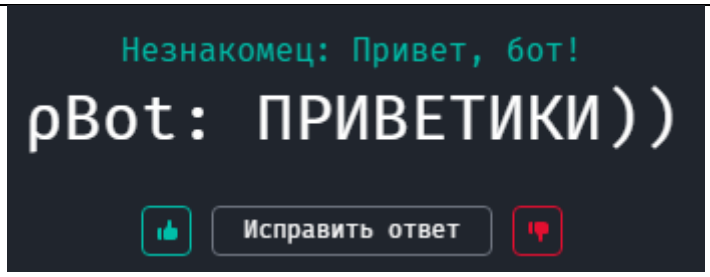
Несмотря на отсутствие коллекций, подобных англоязычным, в сети можно найти некоторое количество диалогов людей с машинами. В данном случае под машинами подразумеваются чат-боты, что привносит в такие диалоги определённую специфику. Во-первых, чат-боты дают возможность исключительно письменной коммуникации, без поддержки голосового ввода. Основное преимущество такого способа общения состоит в отсутствии ошибок распознавания, что облегчает работу по поиску других ошибок. Во-вторых, чат-боты не целеориентированны (*non goal-oriented*), то есть ориентированы на общение, а не на выполнение конкретных задач. С одной стороны, это ограничивает коллекцию данных и делает её менее репрезентативной с точки зрения современной стадии развития диалоговых систем. С другой стороны, в не целеориентированных диалогах больше дискурсивного разнообразия и, следовательно, возможностей для совершения ошибок. В-третьих, важно отметить, что чат-боты могут быть обучены по-разному. Различны могут быть не только методы обучения (см. подробнее об обучении каждого бота в соответствующих пунктах), но и его продолжительность и качество.

Таким образом, записи диалогов с различными чат-ботами полностью отвечают требованиям к данным, необходимым для решения поставленной задачи. Некоторым недостатком этих данных можно считать тот факт, что ныне доступные данные представляют собой записи диалогов, реально полученные 7-10 лет назад, поскольку чат-боты были наиболее популярны в 2007–2010 гг. Сейчас, с одной стороны, создаётся меньше ботов-собеседников (при том, что появляется больше функциональных ботов), а с другой – люди менее активно пользуются чатами. Многие проекты, созданные в то время, в настоящее время поддерживаются слабо или не поддерживаются вообще, однако старые записи иногда доступны. Соответственно, архивы отражают уровень широко доступных технологий (который заметно повысился за последние годы) и, что особенно важно, ожидания пользователей, основанные на доступных технологиях. Тем не менее кажется, что стратегии, которые пользователи использовали при общении с ботами ранее, во многом идентичны тем, которые используются сейчас и, вероятно, будут использоваться в дальнейшем.

Коллекция, на которой основано настоящее исследование, основана на ряде источников, описание каждого из которых представлено ниже.

Чатбот ρBot

Чатбот ρBot (также известный как roBot) находится в открытом доступе в Интернете[3], а также реализован как приложение на iOS и Android. Бот реализован как статистический, т. е. использует для обучения реплики, уже имеющиеся в его базе. Неизвестно, что составляет основу базы знаний бота, однако в настоящее время пользователи могут самостоятельно исправлять ответы бота и добавлять их в базу (см. Таб. 2)



Таб. 2 Диалог с ботом. Реплика бота сопровождается иконками оценки и исправления ответа.

База знаний бота в настоящее время содержит около 325 тыс. реплик[4].

Пользователи активно участвуют не только в пополнении базы, но и в её модерации. Каждый может добавить реплику, и самые активные пользователи получают высокое место в рейтинге. За каждую реплику бота можно голосовать. Реплики, набравшие много положительных голосов, появляются чаще, а реплики, набравшие много отрицательных голосов, удаляются из базы. Исправление ответа бота считается внесением новой реплики.

На сайте чат-бота был размещён архив, содержащий 1000 диалогов[5]. (В настоящее время используется обновлённая версия сайта, где этот архив отсутствует.) Время записи диалогов неизвестно. Длина диалогов составляет от 40 до 500 реплик, что очень велико для диалогов человека с чат-ботом. Утверждается, что все диалоги происходят между ботом и людьми, однако внимательное прочтение некоторых диалогов заставляет в этом усомниться. В новой версии сайта есть раздел «Авто-диалог», в котором бот может вести непрерывный разговор с самим собой. Утверждается, что это помогает отслеживать этапы обучения бота. Не исключено, что подобные данные использовались также в старой версии сайта и попали в архив диалогов.

Архив бота ρBot – самая большая коллекция данных от одного бота в используемом в настоящем исследовании корпусе. Её преимуществом (помимо объёма) является относительно хорошая обученность бота и разнообразие его собеседников. К недостаткам

[3] http://chatbot.tw1.ru/index.htm
[4] По данным на 23 мая 2018 года.
[5] По состоянию на март 2017 года.

можно отнести обилие слов, написанных с нарушениями орфографических норм, что затрудняет работу с текстом.

Чатбот Маришко

Маришко – авторский проект[6], разработка которого началась в 2003 году. В настоящем исследовании были использованы любезно предоставленные разработчиком диалоги ICQ версии бота, записанные в 2006 году.

Маришко, как и ρBot, обучается при помощи фраз, используемых пользователями в диалогах, а также заносимых в базу через специальную форму. Автор подробно описывает этапы развития системы обучения своего бота. Так, бот отправлялся в разные чаты с различной аудиторией, чтобы получить более полные знания о том, какие бывают пользователи. Впоследствии реплики бота взвешивались с целью выделения невежливых и оскорбительных. Кроме того, был проведён ряд опросов, где пользователям было предложено отвечать на распространённые вопросы (например, «что такое любовь»), что также способствовало обучению бота. В настоящий момент база знаний бота содержит около 41 тыс. реплик[7].

Однако несмотря на модерацию, многие фразы содержат слова с орфографическими ошибками и опечатками, а также записи на *олбанском* языке. В целом, корпус диалогов с Маришко похож на корпус диалогов с ρBot.

Коллекция диалогов с ботом Маришко, предоставленная автором в наше распоряжение, составляет 35 диалогов 6 до 170 реплик. Все диалоги происходили между ботом Маришко и реальными пользователями ICQ.

Инфы

Основную часть корпуса, однако, составляют диалоги, отличающиеся от вышеописанных по методу обучения. Это диалоги с так называемыми инфами. Инфы – чат-боты, создаваемые любителями на платформе, предоставленной компанией «Наносемантика»[8]. Термин *инф* происходит от аббревиации *информационный эльф*, поскольку инфы позиционируются компанией как помощники в самых различных делах. Однако до того, как «Наносемантика» стала активно выпускать бизнес-инфов, напоминающих современные целеориентированные диалоговые системы, инфы долгое время и в большом количестве создавались обычными пользователями для повседневного

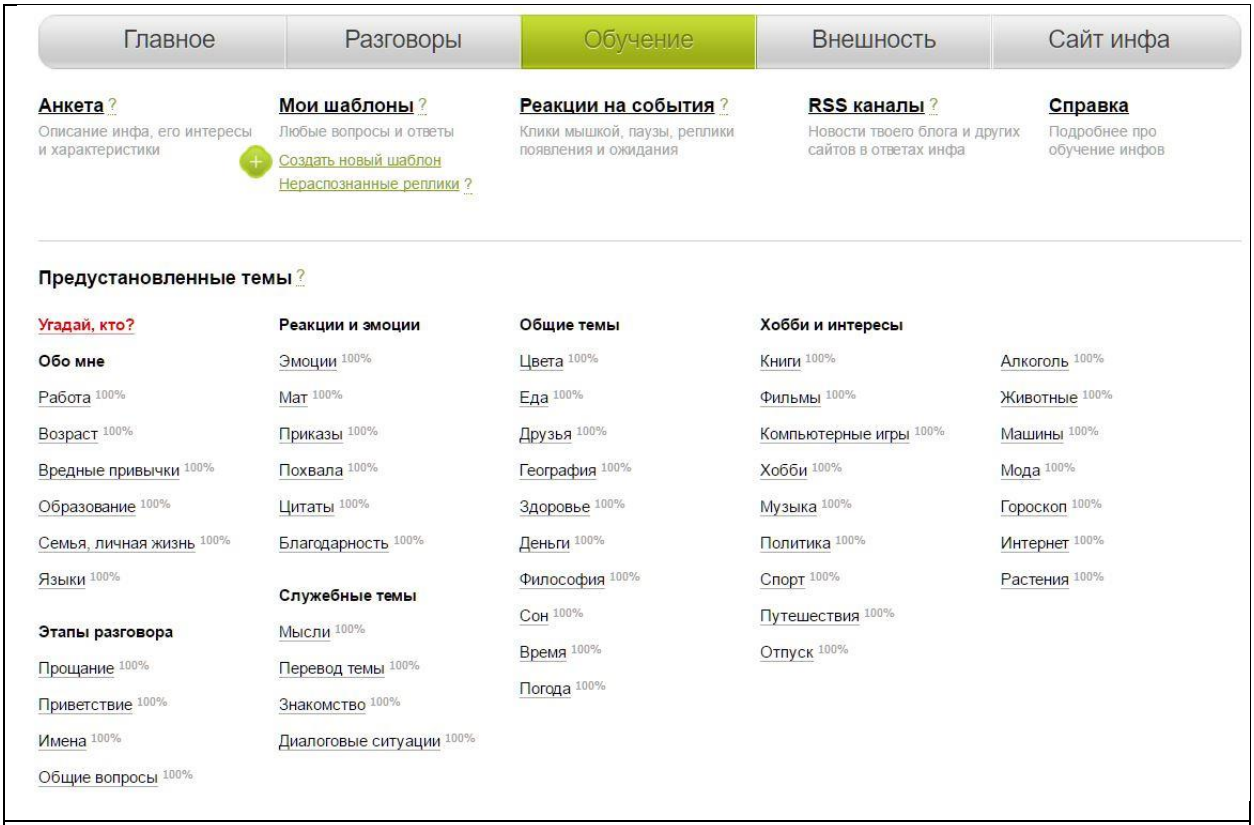
[6] Сайт бота: http://marishko.gorcer.com/index.php?page=about, страница разработчика: https://github.com/gorcer.
[7] По данным на 23 мая 2018 года.
[8] http://iii.ru/

общения. Платформа была популярна в 2009–2010 гг. К 1 декабря 2010 г. на сайте был зарегистрирован один миллион инфов. Впоследствии некоторые инфы были удалены, некоторые заброшены, а некоторые экспортированы на другие сайты. В настоящее время поддержка сервиса инфов осуществляется слабо, и основное внимание компании «Наносемантика» в этой сфере уделено бизнес-инфам[9].

Платформа предоставляет широкие возможности для обучения инфов. Основное обучение происходит за счёт шаблонов. Шаблоном называется пара вопрос-ответ. Тематику шаблонов каждый разработчик инфа определяет самостоятельно, однако компания предлагает набор рекомендованных к заполнению тем, где содержатся наиболее частотные вопросы реплики, без которых трудно вести общение (например, перевод темы или ответ на непонятный вопрос). В процессе обучения пользователи самостоятельно создают шаблоны для своих инфов. Для этого они могут использовать вопросы, уже заданные инфу, или придумывать свои (например, разного рода игры). Страница обучения инфа содержит раздел «нераспознанные реплики» – те реплики, для ответа на которые инфу пришлось использовать один из служебных ответов (например, «я не знаю», «скажи по-другому» и т.п.) В шаблонах поддерживаются элементы регулярных выражений (любое слово), инлайн-словари (возможность выбрать слово в ответе из заданного списка), ссылки на реплики собеседника, передача гиперссылок. Помимо обучения на шаблонах возможно обучение с помощью редактирования ответов, подобно реализованному в ρBot и Маришко. В случае с инфами редактировать ответы может только сам автор инфа в сохранённых диалогах. Тогда отредактированные пары вопрос-ответ добавляются в базу шаблонов.

Один пользователь может создать сколько угодно инфов, что некоторые делали для создания ярких ролевых персонажей. В результате этого появлялись нестандартные ответы на стандартные вопросы, что не давало диалогам развиваться всё время по одному сценарию. Общность набора шаблонов (по крайней мере, стандартного ядра) и разнообразие их наполнения позволяют инфам вести разговоры друг с другом, без участия собеседника-человека. Автор инфа, инициирующего разговор, должен ввести первую реплику, остальные реплики генерируются автоматически. Диалог завершается после 19-ой реплики второго (не инициировавшего разговор) собеседника. Таким образом, длина каждого такого диалога не превышает 48 реплик. В рамках настоящего исследования было принято решение рассматривать такие машинно-машинные диалоги наравне с человеко-машинными.

[9] https://inf.ai/



Таб 3. Тематика диалогов с инфами.

Большинство авторов инфов – подростки и молодёжь, что определяет круг обсуждаемых тем. Инфы обсуждают увлечения и интересы, многие готовы поговорить о кино и литературе, путешествиях, субкультурах и интересных фактах.

Диалоги с инфами доступны только авторам инфов, участвующих в диалоге, и не размещаются в едином архиве. Более того, срок хранения диалогов на сервере ограничен, поэтому невозможно получить длинную историю диалогов от одного инфа. В связи с этим было принято решение обратиться к форуму сайта.

Форум инфов

На сайте есть форум для общения разработчиков инфов, содержащий Цитатник инфов, то есть избранные фрагменты диалогов. Материалы этой ветки легли в основу коллекции диалогов с инфами. Форум содержит как человеко-машинные диалоги (диалоги инфов с реальными пользователями), так и машинно-машинные (разговоры инфов с инфами). Как правило, автор заводит тему для публикации диалогов своего инфа, и выкладывает там время от времени какие-то материалы, а другие пользователи выражают своё отношение к этим материалам. Также люди-собеседники, поговорив с инфом, могут выложить понравившийся диалог в тему, посвящённую разговорам с этим инфом.

Поскольку на форуме как правило публикуются фрагменты разговоров, длина диалогов невелика, от 2 до 20 реплик. В корпусе для настоящего исследования содержится около 2500 диалогов, извлечённых из форума.

Такие диалоги, как правило, не содержат традиционных для разговора приветствий и прощаний. Тематика диалогов разнообразна, поскольку в выборку чаще всего попадают понравившиеся пользователям индивидуальные шаблоны инфов или интересные авторские ответы на стандартные вопросы. Значительную часть таких диалогов составляют забавные ситуации непонимания инфом собеседника, что представляет особую ценность для настоящего исследования, поскольку в таких диалогах содержатся искомые реплики. Также частотны случаи, когда инф остроумно отвечает на реплику, свидетельствующую о неудовлетворённости его собеседника (см. Таб. 4).

Пример 1
X: ученые всего мира доказали, что теплее станет только в тундре, пар заслонит солнце - и мы все замерзнем! теперь ты знаешь, что делать? Y: снимать штаны, и бегать! X: лучше не надо. Y: я не поняла.

Таб 4. Пример диалогов, размещаемых на форуме

Основным недостатком этой коллекции оказывается её нерепрезентативность. Выборка диалогов не сплошная, и критерии, которыми руководствовались посетители форума, размещая те или иные фрагменты, не известны. Это обстоятельство может иметь влияние на количественные характеристики корпуса. Специально выбранные фрагменты могут увеличить долю искомых реплик по сравнению с их долей в диалогах, полученных сплошным способом. Впрочем, следует сказать, что некоторая фрагментарность присуща всем коллекциям, вошедшим в состав корпуса для настоящего исследования. Тем не менее, важным преимуществом коллекции форума инфов остаётся тот факт, что многие фрагменты показались пользователям интересными или смешными именно за счёт содержащихся в них ошибок, и поэтому этот источник представляет большую ценность для настоящего исследования.

Несмотря на труднодоступность полной выборки диалогов от инфов, некоторые разработчики предоставили такие данные в наше распоряжение.

Инф Кармен

Инф Кармен – частный инф, созданный в 2009 году. Инф давно не обновлялся и содержит не очень много пользовательских шаблонов, однако стандартная библиотека заполнена на 100% и содержит некоторые реплики, отражающие личное мнение автора по ряду вопросов. Ввиду ограничения на срок хранения данных, не удалось получить разговоры этого инфа за тот период, когда инфы активно использовались. Тем не менее, разработчик инфа предоставил ряд диалогов между двумя инфами (один из которых – Кармен), собранных в 2016–2017 гг. Эти диалоги имеют стандартную длину по 19 реплик с каждой стороны. Невысокая степень персонализации инфов в сочетании с нечеловеческой стратегией ведения диалога с другой стороны иногда приводит к неожиданным эффектам (Таб. 5). Таким образом, коллекция диалогов с инфом Кармен оказалась весьма насыщенной репликами, представляющими интерес для настоящего исследования.

Пример 1
X: ты любишь мороженое? Y: упс, я потерял ответ на твой вопрос или даже сломал, сорри... X: да ничего =) Y: а у тебя был какой-нибудь сон, который никак нельзя было забыть? и о чём он был? X: вся наша жизнь может оказаться сном... Y: опять! опять разрыв связи был. чортовы связисты! X: хм... Y: а что тут непонятного? X: не поняла вопроса :(Y: не страшно, поймём друг друга со временем) у тебя есть брат или сестра? X: может, не стоит обсуждать личные темы?

Таб. 5. Пример диалога с Инфом Кармен (собеседник X)

Инф Aplgallery

Ещё одно небольшое собрание диалогов было предоставлено разработчиком инфа Aplgallery. Эти разговоры в основном происходят между двумя инфами (собраны в 2016-2017 гг.), хотя есть и разговоры с людьми (собраны в 2016-2017 гг. и найдены в архивах за 2010 г). Длина диалогов составляет от 6 до 60 реплик. Диалоги отражают в основном поверхностные разговоры при знакомстве (small talk), также обнаруживается хорошая проработка шаблонов темы спорта. Как и инф Кармен, инф Aplgallery имеет нестандартные ответы на стандартные вопросы, что делает общение с ним непредсказуемым, особенно в режиме машинно-машинного диалога. Как отметил разработчик инфа в личном сообщении, инф давно не обучался, и уровень его знаний такой же, каким был 7 лет назад. Это позволяет интерпретировать диалоги с Aplgallery наравне с диалогами с другими инфами.

Благодаря коллекциям данных от инфов Кармен и Aplgallery данные о ошибках и примечательных случаях в разговорах с инфами, полученные из материалов форума, подкрепляются некоторым количеством «обычных», не заслуживающих отдельного упоминания в «Цитатнике» диалогов, что делает весь корпус несколько более приближенным к ситуации, когда все диалоги были бы получены методом сплошной выборки.

Инф Alisa_fpml

Многие инфы, созданные на платформе, прдоставленной компанией «Наносемантика» были перенесены на другие сайты. Некоторые инфы были превращены в чат-боты для социальных сетей. Так, разработчик инфа Alisa_fpml перенесла своего бота в сеть Вконтакте, где он общается с реальными пользователями вместо своей хозяйки. Обучение бота продолжается, и автор инфа создаёт новые шаблонные вопросно-ответные пары на основании общения бота в новом окружении.

Выборка, предоставленная Алисой, не сплошная, однако значительно менее фрагментарная, чем материалы форума. Автором были выбраны диалоги, содержащие такую информацию, которую автор посчитала возможным сообщить. В итоге были получены материалы диалогов с 10 пользователями, каждый из которых содержит историю сообщений за несколько недель и имеет длину более 100 реплик. Тематика разговоров с Алисой разнообразна, однако по большей части касается отношении. Общение кажется очень похожим на человеческое и случаи сбоев нечасты.

Сравнение источников

Вообще, материалы, полученные от всех ботов, большей частью содержат диалоги на простые бытовые темы и направлены на установление контакта с собеседником и получение некоторой информации о нём, равно как и сообщение определённой информации о себе. Успешность бота в таком случае зависит от проработанности его системы понятий. Онтологическая архитектура в чистом виде не применялась ни в одном из рассмотренных ботов. Система обучения инфа предполагает развитие некоторой иерархии шаблонов, связки реплик между собой, и перехода с одной темы на другую, что в некотором роде сродни созданию базы знаний (а не только базы реплик). Для ботов, основанных исключительно на добавленных пользователями репликах (ρBot и Маришко) проработанность темы обусловлена частотностью этой темы в разговорах различных пользователей.

Эти особенности коррелируют с представленностью различных типов реплик, свидетельствующих об отрицательной оценке собеседника, в диалогах различной архитектуры. В системах, основанных на шаблонах, чаще встречаются конкретизирующие реплики: «скажи по-другому», «можешь использовать другое слово?» и подобные. В системах, основанных на репликах пользователей, чаще используются характеризующие реплики общего характера: «дура», «сумасшедшая» и подобные.

Сравнение всех источников материала можно видеть в Таб. 6.

	roBot	Инфы	Частные боты
Автор	?	около 100 авторов	4 автора
Архитектура	статистический	фреймы, правила	правила
Способ взаимодействия	интерфейс на сайте, приложение на iOS и Android	интерфейс на сайте	социальные сети
Объём корпуса	1000 диалогов	около 2500 диалогов	около 70 диалогов
Длина диалога	от 40 до 500 реплик	от 2 до 20 реплик	от 6 до 200 реплик
Тематика	отношения	увлечения, интересные факты	увлечения, отношения, путешествия, спорт

Таб. 6. Сравнение источников данных

Первичная обработка материала

Материал из всех источников был получен в формате .txt или в формате .html. В последствии из всех файлов были извлечены только тексты диалогов, удалены цитаты и повторения. Все диалоги были сконвертированы в файлы формата .txt в кодировке UTF-8.

Первичная обработка материалов включала в себя в основном типографическую работу с целью нормировки данных. Так, приведение всех слов к нижнему регистру и отделение знаков препинания от последовательностей буквенных символов позволило несколько снизить количество вариантов написания одного и того же слова. С этой же целью последовательности из трёх и более одинаковых букв были сведены к одной букве (например, *бооооооооольшой* было превращено в *большой*). Однако эти меры всё равно не позволяют справиться с различными орфографическими ошибками и опечатками.

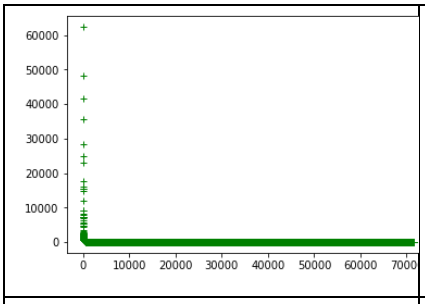
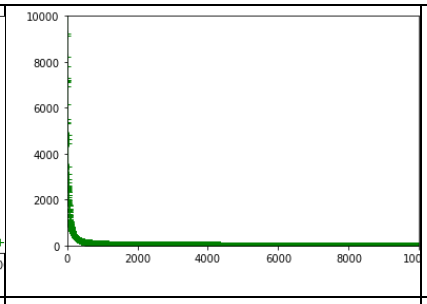
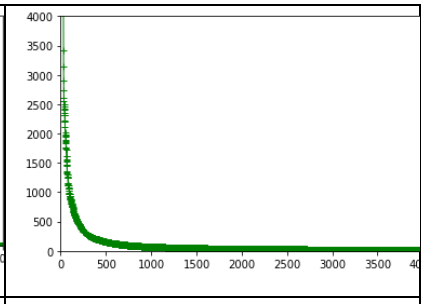
Нормировка данных проводилась таким образом, чтобы не потерять информации, которая могла бы иметь влияние на автоматическую обработку содержания предложений. Поэтому знаки препинания были отделены от слов, но не удалены полностью. То же касается и лемматизации. Несмотря на то, что лемматизация позволила бы уменьшить вариативность форм, её применение привело бы к потере грамматической информации, что кажется недопустимым при анализе предложений, а не отдельных слов.

Последним этапом стала унификация имён собеседников. Для решения поставленной задачи не важно, кто именно произносит ту или иную реплику, важно лишь отследить, что последовательные реплики произносятся разными собеседниками. Если две последовательные реплики в диалоге исходили от одного источника, то эти две реплики объединялись в одну. Для удобства восприятия диалогов, репликам были приписаны условные коды собеседников X и Y.

В результате объём корпуса составил 288 209 реплик, состоящих из 2 115 404 токенов (последовательностей символов между двумя пробелами, в том числе знаков препинания). Средняя длина реплики, таким образом, составляет 7.4 слова.

Среди всего количества токенов можно найти 71 574 уникальных. Получается, каждое слово повторяется в корпусе в среднем 29 раз (что, конечно, даёт весьма приблизительную оценку разнообразия корпуса, поскольку частотность слов очень различна). Так как лемматизация не проводилась, это число включает в себя все различные формы одного и того же слова, включая написанные с графическими или орфографическими нарушениями.

Тем не менее, несмотря на нестандартность и ненормированность данных, они в целом отражают картину, характерную для данных естественного человеческого языка, что доказывает, в частности, соответствие распределению Ципфа (см. Таб. 7).

		
А) все данные	Б) фрагмент графика: до 10 000 по каждой из осей	В) фрагмент графика: до 4 000 по каждой из осей

Таб. 7. Распределение уникальных слов в корпусе. По оси ординат – частота на миллион (ipm), по оси абсцисс – ранг по частоте.

Результаты предварительной обработки были сочтены удовлетворительными для проведения дальнейшего исследования.

Раздел 3: Создание признакового пространства

Постановка проблемы

Автоматическая обработка текстовых данных в их исходном виде невозможна. Для того, чтобы компьютерная программа могла осуществлять различные операции с данными, они должны быть представлены в численном виде.

Для представления текстовых данных в численном виде могут использоваться различные методы. Для некоторых задач оказывается полезным использование hash-функции, имеющей уникальное значение для каждого уникального текстового отрезка (в связи с этой технологией нельзя не упомянуть новую работу [Kallmeyer et al. 2017], где результативность представления слов в виде векторов посредством hash-функций оказывается выше, чем у алгоритма word2vec). По значению hash-функции крайне затруднительно определить особенности строения или содержания текста, поэтому такой метод удобен только в том случае, когда каждый текст важен в рамках задачи как единая сущность, а его отдельные характеристики значения не имеют.

Для анализа текста как набора некоторых характеристик представляется значительно более эффективным представить текст через интересующие исследователя характеристики. Характеристики могут касаться любого уровня построения прредложения, например, количество букв, отношение количества служебных морфем к количеству корней, номер слова, обозначающего подлежащее и т. п. Однако поиск числовых характеристик, осмысленно отражающих содержание текста, может оказаться отдельной непростой задачей для исследования.

В последнее время в компьютерной лингвистике всё большее распространение получают методы дистрибутивной семантики. Они основываются на положениях о том, что контекстное употребление слова задаёт его значение и что слова, встречающиеся в похожих контекстах, имеют похожие значения ([Harris 1954]) Иными словами, похожие слова употребляются в похожих контекстах, а разные – в разных. При таком подходе каждый текст представляется как набор контекстов.

Традиционные методы дистрибутивной семантики предполагают отмечать, как часто каждое слово встречается в заданной окрестности других слов. Поскольку оценивать близость каждого слова к каждому очень затратно, обычно выбирают некоторое количество наиболее частотных слов, и для каждого слова оценивают частоту появления рядом с каждым из выбранных. Получается, что каждое слово задаётся набором чисел. Последовательность этих чисел называется вектором слова, а количество чисел в этой последовательности – размерностью вектора. Совокупность многомерных векторов, кодирующих признаки, называется признаковым пространством. Подобная генерация

признаков значительно проще, чем осуществляемая исключительно силами исследователя. Однако такой метод сопряжён с рядом недостатков. Главное неудобство заключается в том, что не все слова находятся рядом со всеми, а потому векторы содержат много нулей, что не очень удобно для их сравнения.

В 2013 г. Томаш Миколов представил нейросетевой алгоритм, самостоятельно выбирающий признаки для характеристики слов ([Mikolov et al. 2013]). Этот алгоритм был сначала применён к словам, а затем модифицирован для обработки текстов любой длины ([Mikolov & Le 2014]). Алгоритм неоднократно применялся в различных исследованиях, в том числе и на материале русского языка (среди работ последних лет выделяются исследования [Maslova & Potapov 2017] по определению тональности и [Mescheryakova 2017] по классификации текстов). Недостатком этого метода оказывается невозможность контроля над ним со стороны исследователя. В отличие от случаев, когда текст представляется в виде имеющих физический смысл характеристик, и каждое число в векторе имеет объяснимое отношение к реальности, векторы, созданные нейросетью, не интерпретируемы. Каждое число ничего не значит, однако сходство или различие векторов в целом можно определять математическими методами (чаще всего используется мера косинусной близости: чем косинус между векторами больше, тем они ближе).

Для того, чтобы рассмотреть новые в научном обиходе данные с разных сторон и заметить как можно больше их особенностей, было создано три признаковых пространства различными методами. Поскольку каждый метод имеет свои достоинства и недостатки, трудно оценить на этапе планирования исследования, какое именно признаковое пространство окажется наиболее пригодным для анализа. Все признаковые пространства используются на всех дальнейших этапах исследования. Каждый метод ниже описан в подробностях. Представляется, что сложность и, напротив, успешный опыт применения описываемых методов могут быть интересными как некоторые промежуточные практические результаты настоящего исследования.

Раздел состоит из трёх подразделов, каждый из которых посвящён описанию одного метода создания признакового пространства. В каждом подразделе даётся описание алгоритма и параметры его настройки, затем приводятся и обсуждаются результаты применения метода к имеющимся данным.

Автоматический метод: doc2vec

Первое признаковое пространство было создано автоматически с использованием встроенных средств алгоритма doc2vec (далее такое пространство будет называться A-пространством). Как уже отмечалось, алгоритм doc2vec построен аналогично алгоритму word2vec, т.е. рассчитывает вероятность появления слов в похожих контекстах.

Применимость алгоритма doc2vec к текстам различной длины обеспечивается за счёт создания специального вектора документа. Технически это реализовано как идентификационный номер текста в особом формате, позволяющий отличить этот номер от обычных чисел внутри текста. Этот номер позволяет алгоритму определить длину текста и нормировать длину вектора соответственно. В результате, тексты разной длины оказываются закодированы векторами, которые можно сравнивать с помощью косинусной меры или обрабатывать иными способами (см. [Mikolov & Le 2014; Lau & Baldwin 2016]).

Главным достоинством алгоритма doc2vec является простота его использования. Все этапы тренировки уже реализованы внутри библиотеки, и остаётся лишь подобрать оптимальные параметры для конкретного корпуса данных.

В рамках настоящего исследования было натренировано несколько десятков моделей с различными параметрами. Каждая модель оценивалась нами на основании того, какие ближайшие реплики-синонимы она подобрала для некоторого набора диагностических реплик, а также как она оценивала близость двух высказываний (которую мы полагали почти равной единице – высказывания отличались на одно слово). Такой метод оценки, хотя и не опирается на формальные метрики, позволяет, благодаря задействованию экспертного мнения, получить быстрое и адекватное впечатление о качестве модели на основании небольшого количества данных.

Также при обучении и качественной оценке различных моделей было сделано два математических наблюдения, которые косвенно свидетельствовали о качестве модели (см. Таб. 8). Первый эффект заключается в том, что плохо обученные модели имеют в целом низкие значения косинусных мер. То есть, в том случае, когда мы бы ожидали увидеть значение, близкое к 1, получается значение, например, около 0.7 (в разных моделях достигается разное значение, но оно в любом случае невелико). При этом относительное распределение косинусных мер может сохраняться, то есть это значение окажется наибольшим среди всех имеющихся (что мы и ожидали бы от гипотетического значения 1 в таком случае). Как кажется, такое явление может наблюдаться в тех случаях, когда модель переобучается. Под переобучением понимается явление, при котором алгоритм извлекает слишком много признаков объектов и считает их значимыми (что неверно), в результате считая непохожим то, что не следует.

==========радость======	
наслаждаться	0.69
одинокий	0.68
сейчас	0.68
впервые	0.67
влюбленный	0.66
выплескивать	0.66
уходить	0.66
бесчестие	0.65
напоминать	0.65
поцеловать	0.65
какой	0.65
написать	0.64
вызывать	0.64
казаться	0.64
теперь	0.64
любовь	0.64
удовольствие	0.64
встречать	0.64
чуждый	0.64
конечно	0.63

Таб. 8. Пример плохо обученной модели

Второй эффект заключается в том, что в плохо обученных моделях различные синонимы имеют одну и ту же косинусную меру сходства с заданным словом. Иными словами, даже среди десяти ближайших синонимов нет строгого ранжирования от самого близкого к наименее близкому (в примере в Таб. 8 первые 10 синонимов имеют всего 5 различных значений меры косинусной близости). Вместо этого выделяется некая группа одинаково близких слов. Можно было бы предположить, что, действительно, произошло такое совпадение, что несколько слов имеют абсолютно одинаковое векторное представление. Тогда косинусная мера их близости была бы равна 1. Нетрудно убедиться, что в случае подобного недостатка обучения это не так. В этом случае, видимо, речь идёт о недообучении: модель извлекает недостаточно данных, по которым она могла бы охарактеризовать предложения, и поэтому не делает различия между некоторыми высказываниями.

Для тренировки моделей в рамках настоящего исследования использовался пакет word2vec для Python[10]. Метод тренировки моделей, реализованный в этом пакете, позволяет настраивать значительное количество параметров. В данной работе нам бы не хотелось слишком вдаваться в технические и математические подробности того, почему эти параметры доступны и как они гипотетически должны влиять на результаты тренировки (такая информация доступна в документации к пакету и статьях об алгоритме doc2vec). Мы бы хотели остановиться только на тех параметрах, с которыми действительно работали, и

[10] Репозиторий разработчика: https://github.com/danielfrg/word2vec (10.06.2018)

перечислить некоторые сложности их настройки. Мы также воздерживаемся от цитирования результатов неудачных моделей.

Традиционно считается, что именно **размерность** является ключевым параметром любого признакового пространства. Обычно замечают, что чем она больше, тем больше признаков можно уловить и, следовательно, тем точнее уловить близость элементов. Это вступает в противоречие с требованиями к алгоритму: чтобы он работал быстро, а выходные данные не занимали много памяти, оптимально выбирать минимальную размерность. Исходя из этого противоречия, можно сформулировать идеальное решение: необходимо найти модель с такой размерностью, что при увеличении количества измерений значительного улучшения качества не происходит, и взять её как самую экономичную при прочих равных. Важной проблемой размерности оказывается переобучение: слишком большое количество признаков приводит к ухудшению результатов, поскольку алгоритм старается уловить несуществующие различия.

В рамках исследования были созданы модели с размерностью от 30 до 2000. Лучшие результаты были продемонстрированы моделью с размерностью 100.

Вторым по важности параметром чаще всего называют **ширину окна**. Под этим термином понимают окрестность, в которой для заданного слова ищутся слова, определяющие его контекст. То есть, если задана ширина окна 1, будут рассмотрены только слова, непосредственно предшествующие заданному или непосредственно следующие за ним. Бытует мнение, что чем короче предложения в корпусе, чем уже должно быть окно, поскольку на большом расстоянии от заданного слова уже может оказаться не связанное с ним предложение. В художественных произведениях, изобилующих сложными периодами, напротив, оправдан выбор широкого окна, способного уловить связи очень отдалённых слов. Однако представляется, что этот параметр зависит также от языка и типа текста[11].

Поскольку реплики чат-ботов коротки, было предположение искать достаточно узкое окно. Тем не менее, натренированы модели с шириной окна от 1 до 15, и лучшие результаты были продемонстрированы моделью с шириной окна 4.

Алгоритм word2vec позволяет использовать два типа грамматик при тренировке моделей. Грамматика CBOW (continuous bag of words) принимает на вход контекст слова и предсказывает вероятность появления слова в нём. Грамматика Skip gram, напротив, предсказывает наиболее вероятный контекст при данном слове.

[11] Этой проблеме был посвящён доклад “Shall we or shall we not? Stop words in distributional semantics from a comparative perspective” на конференции Rencontres Jeunes Chercheurs 2018 “Des données à la théorie” 31 мая 2018 года в Париже: http://www.univ-paris3.fr/le-programme-des-rjc-2018-362276.kjsp?RH=1416846462568 (10.06.2018)

Разные модели были натренированы с каждым из двух типов грамматики, и тип ‘cbow’ стабильно показывает более осмысленные результаты, чем ‘skipgram’.

Зачастую в больших корпусах очень редкие слова удаляются путём установления порога на **минимальное число вхождений слова.** Предполагается, что слово, которое встречается в корпусе очень редко, не представляет интереса для исследования, но может «испортить» своим присутствием векторы других слов.

Поскольку наш корпус невелик, во всех моделях минимальным порогом было одно вхождение, то есть оценивались все слова.

Такие параметры, как **количество итераций** и **критерий сглаживания**, хотя и принимали разное значение в разных моделях, не продемонстрировали существенного влияния на результаты, поэтому в итоге использовались стандартные настройки этих параметров.

Принимая во внимание успехи алгоритма word2vec, основывающегося на нейросетевой архитектуре, можно было бы предполагать, что и алгоритм doc2vec, унаследовавший эту архитектуру, будет эффективным. В данном случае под эффективностью понимается способность находить сходство между семантически близкими текстами и не находить его между семантически далёкими.

Ниже представлены некоторые высказывания и ближайшие к ним результаты. Высказывания выбраны из корпуса так, чтобы можно было оценить работу алгоритма как с эмоциональной, так и с неэмоциональной лексикой различной тематики. Ниже рассматриваются четыре тестовых высказывания и ближайшие к ним (по данным автоматического алгоритма создания признакового пространства) реплики.

Все представленные ниже результаты получены от модели, продемонстрировавшей лучшие результаты на этапе тестирования. Авторское написание сохранено.

Пример 1. Высказывание со словами, отрицательно характеризующими собеседника.

Пример (предложение 0) содержит эмоционально окрашенные слова: *дурак, глупые, глупышки*. Первое из них встречается в трёх из пяти ближайших высказываний. При этом только в предложении 4 оно используется в той же иллокутивной ситуации, что и в примере. Интересно, что в предложениях 2 и 5 словом повторяется используемое в примере слово *давно*, которое, как кажется, не является ни семантически, ни синтаксически выделенным.

0	ты дурак . я давно замечала что инфы - глупые . зачем моей сестре нужны были эти глупышки
1	ты эт уже писал дурак
2	и давно ? ты это не запускай , сходи к врачу пока не поздно !
3	што я дурак
4	ты круглый дурак
5	я давно не проверялась

Таб. 9. Ближайшие синонимы Примера 1 в А-пространстве

Пример 2. Высказывание со словами, выражающими положительные эмоции.

Пример (предложение 0) содержит положительно эмоционально окрашенные слова: *замечательно, рад*. Предложения 1 и 2 содержат крупный фрагмент высказывания *это просто замечательно*, в точности повторяющий фрагмент примера. Важно отметить, что во всех предложениях 0-2 этот фрагмент стоит в начале. В этих случаях можно говорить о том, что сочетание слов было воспринято как единая коллокация. Примеры 4 и 5 интересны тем, что в них словосочетание *это просто замечательно* рассматривается как конструкция в традиционном понимании (вслед за [Kay & Fillmore 1999]). В предложении 4 (предикативное) наречие *замечательно* заменяется на *ужасно*, что действительно возможно в рамках конструкции. В предложениях 3 и 5 происходит более существенная трансформация: наречие заменяется на существительное, в предложении 5 также добавляется отрицание. Получается, что в случае Примера 2 алгоритм смог уловить конструкцию, часть слов которой константна, и найти другие случаи её появления в корпусе.

0	это просто замечательно ! рад за тебя .
1	это просто замечательно у тебя ещё столько интересного впереди . а ты в каком классе учишся ?
2	это просто замечательно . и кто у тебя - кошечка или котяра ? или толпа ?
3	да это просто злодейство
4	это так ужасно ! не могу , ой , не могу сказать !
5	философия - это не просто наука , это стиль жизни

Таб. 10. Ближайшие синонимы Примера 2 в А-пространстве

Пример 3. Высказывание, не содержащее эмоциональной лексики, употребляющееся в определённых ситуациях.

Этот пример призван проверить, способен ли алгоритм уловить конструкцию, а реализующуюся на уровне единиц больших, чем простые предложения. Для этого был выбран пример с формулой приветствия.

Как видно, в предложениях 1, 2 и 5 повторяется имеющееся в примере слово *здравствуйте*. Вместо предложения представиться используется формула представления,

т.е. конструкция не соблюдается. В предложениях 3 и 4 наследуется конструкция «глагол в повелительном наклонении + *пожалуйста*». Однако в предложении 3 не сохраняется иллокутивная ситуация приветствия.

0	здравствуйте ! представьтесь пожалуйста !
1	здравствуйте! я - таисия васильевна!
2	здравствуйте! я наводчик танка №098 виктор многострелов. поговорим?
3	подождите пожалуйста , больные в четвертой палате разбушевались и требуют лучшего сервиса !
4	пожалуйста представьтесь системе помощи клиентам божьей обители . если вы уже являетесь клиентом - нажмите 0 если вы впервые - нажмите
5	здравствуйте! хотите в отпуск?

Таб. 11. Ближайшие синонимы Примера 3 в А-пространстве

Пример 4. Высказывание, не содержащее эмоционально окрашенной лексики и каких-либо конструкций.

Методологически верным было бы найти высказывание, не содержащее конструкций, и посмотреть, как оно обрабатывается. Однако сделать это затруднительно, поскольку в той или иной степени конструкцией можно считать едва ли не любое сочетание слов, особенно в языке с богатой морфологией. Поэтому было принято решение взять достаточно длинное высказывание, чтобы проследить, какие именно его части алгоритм посчитает коллокациями или конструкциями.

В Таб. 12. видно, что в примере 4 алгоритм нашёл коллокацию *со мной*. В предложениях 2 и 5 она употребляется как предложно-падежная группа, зависящая от глагола. В предложениях 1, 3 и 4 она является частью большей конструкции «*со мной всё* + наречие». В предложении 1 в качестве наречия выступает группа *в порядке*. Она также повторяется в предложении 3 в другом синтаксическом окружении.

0	со мной все понятно , я читатель детективов ищущий носочки . а кто ты по профессии ?
1	со мной все в порядке ! это что с тобой ?
2	поедем со мной на дачу , я знаю , ты любишь копаться в земле
3	со мной все отлично , можете не беспокоиться . и все - все в порядке
4	со мной все хорошо , а с тобой ?
5	поговори со мной

Таб. 12. Ближайшие синонимы Примера 4 в А-пространстве

На первый взгляд может показаться, что алгоритм прекрасно справляется с задачей. Однако нетрудно заметить, что все найденные синонимы содержат хотя бы одно слово, присутствующее в высказывании-запросе. На основании этого можно сделать

предположение, что алгоритм doc2vec, несмотря на встроенные механизмы нормализации всё же остаётся крепко привязанным к векторным представлениям отдельных слов. То есть, семантическая близость учитывается алгоритмом значительно меньше, чем лексическое сходство.

Поскольку алгоритм doc2vec проявил недостаточно чувствительности к семантике высказываний, появилась необходимость в создании другого, более чувствительного алгоритма.

Контролируемый метод

Автоматическому неконтролируемому методу векторизации на основе алгоритма doc2vec мы решили сопоставить метод, основанный на понятных содержательных критериях. С одной стороны, использование метода с интерпретируемыми признаками может позволить получить результаты, интересные для понимания проблематики исследования. При наличии интерпретируемого метода можно будет выявить отдельные характеристики, являющиеся более или менее значимыми отличиями реплик, свидетельствующих об отрицательной оценке собеседника, от других высказываний.

С другой стороны, применение такого метода необходимо с методологической точки зрения. При сопоставлении результатов применения контролируемого метода с результатами применения неконтролируемого можно будет оценить эффективность каждого из них и понять целесообразность использования каждого из методов для решения подобных задач.

Однако, как уже отмечалось, самостоятельное извлечение большого количества признаков, смысл которых можно интерпретировать с лингвистической точки зрения, очень трудоёмко. Поскольку извлечение признаков не является главной целью настоящего исследования, было принято решение рассмотреть исследуемые высказывания только в одном аспекте их содержания, а именно с точки зрения тональности. Под тональностью в русскоязычной компьютерной лингвистике традиционно понимается эмоциональная оценка текста. Если объём текста превышает одно слово, тональность оценивается на основании входящих в него слов. Тональность отдельных слов не оценивается, а задаётся, например, в специальных (тональных) словарях.

Тональность показалась интересным аспектом для исследования по двум причинам. Во-первых, некоторый набор характеристик по тональности уже задан: слова могут быть либо тонально окрашенными, либо нет, а окрашенные слова могут быть либо положительными, либо отрицательными. (Для сравнения: применение синтаксических категорий, например, «предложение либо сложное, либо простое, а если сложное, то либо сложносочинённое либо сложноподчинённое» требует значительно больше как усилий со стороны исследователя, так и вычислительных мощностей.) Во-вторых, тональность может быть интерпретирована без дополнительных умозаключений. Гипотеза относительно тональности достаточно проста и интуитивно кажется, что во многом верна в применении к исследуемому материалу: если в высказывании много отрицательно окрашенных слов, скорее всего, в этом слове присутствует отрицательная оценка собеседника (Продолжая сравнение с синтаксическим аспектом, можно отметить, что факт сложноподчинённости предложения ещё ничего не говорит о том, как настроен собеседник). Некоторым

недостатком оказывается то, что в русском языке немало многозначных слов, которые могут использоваться как в положительном, так и в отрицательном смысле. Кроме того, нередки случаи иронии, при которой слово должно восприниматься в противоположном значении (например: *Я ключи дома забыла, вот умная*). Тем не менее, именно тональность рассматривается как важная характеристика реплик, свидетельствующих о пользовательской оценке, в литературе (например, [Gamon 2004])Всё вышеперечисленное привело к выбору тональности в качестве аспекта, по которому диалоговые реплики были охарактеризованы с целью получения векторов признаков.

Несмотря на то, что оценка тональности является одним из достаточно популярных направлений компьютерной лингвистики (особенно в сфере применения к различным коммерческим проектам), разнообразие предлагаемых готовых решений невелико (небольшой обзор представлен в [Меньшиков, Кудрявцев 2012], однако немало решений интегрированы в различные сайты без указания на источники). Некоторое недоумение вызывает тот факт, что все доступные решения в качестве ответа выдают бинарную оценку: текст либо позитивный, либо негативный (во многих случаях, впрочем, оценка тернарная: позитивный, негативный или нейтральный). Ни один из продуктов не предлагает *оценить*, *насколько* текст позитивный или негативный, то есть не классифицировать его, а охарактеризовать. Некоторые решения предлагают выделять отдельные позитивно и негативно окрашенные фрагменты текста, справедливо полагая, что в тексте значительного объёма могут встретиться фрагменты с разной тональностью, однако это всё равно не выходит за рамки задачи классификации текста.

Очевидно, что классифицирующий подход неприменим для задачи извлечения признаков, поскольку в качестве ответа выдаётся не вектор, а одно число. По этой причине мы вынуждены были отказаться от идеи применить готовое решение к нашему материалу и разработать собственный метод извлечения содержательных характеристик на основе тональности.

Искомый характеризующий подход отчасти реализован в рамках открытой библиотеки Polyglot[12]. На основе [Chen & Skiena 2014] был создан алгоритм, приписывающий оценку отдельным словам, а затем вычисляющий среднюю тональность текста. Авторы статьи создали тональные словари для 136 языков, используя англоязычные ресурсы в качестве исходных данных и расширяя данные за счёт средств машинного перевода и алгоритмов распространения графа (graph propagation algorithm). При этом оценка каждого слова на любом языке выбирается из трёх значений (-1 для негативных

[12] Разработчик: Rami Al-Rfou, репозиторий разработчика: https://github.com/aboSamoor/polyglot (дата обращения: 10.06.2018)

слов, 0 для нейтральных слов, 1 для позитивных слов), а оценка текста может оказаться любым действительным числом в границах от -1 до 1. Такой формат ответа позволяет по крайней мере ранжировать высказывания, что уже даёт больше информации, чем простая классификация, однако всё ещё не является интерпретируемым набором характеристик текста.

За основу алгоритма векторизации можно было бы взять результаты первого этапа алгоритма библиотеки Polyglot, то есть оценки тональности для каждого слова. Однако в таком случае вектор для каждого высказывания был бы иной размерности, поскольку количество слов в высказываниях неодинаково. Можно было бы создать вектор единой размерности, соответствующей максимальной длине высказывания, однако нули в «пустых» измерениях значительно затемнили бы данные. Кроме того, такие векторы бы показывали не просто тональность текста, а её линейное распределение, поскольку номер измерения соответствовал бы номеру слова в предложении. В русском языке порядок слов в предложении может быть крайне разнообразным, а эмоционально окрашенным может быть слово любой части речи, поэтому знание позиции эмоционально окрашенного слова в предложении не сообщает никакой важной информации об этом предложении. Ввиду этого необходимо было создать метод, характеризующий тональность предложения в целом.

Как представляется, значимым может оказаться абсолютное количество эмоционально окрашенных слов в высказывании, а также доля таких слов в высказывании. При этом было бы интересно посчитать как можно больше различных соотношений, чтобы впоследствии определить, насколько характерным каждое из низ оказывается для реплик, свидетельствующих об отрицательной оценке собеседника.

Для определения тональности отдельных слов был взят крупнейший тональный словарь, созданный на материале русского языка – RuSentiLex ([Лукашевич, Левчик 2016]) Этот словарь был устроен следующим образом. В нём содержатся слова и сочетания слов (вокабулы) в той форме, в которой они были бы помещены в традиционный словарь (главное слово в начальной форме, остальные в формах, продиктованных правилами согласования и управления), каждое из которых снабжено следующей информацией:

- Частеречная принадлежность (для словосочетаний введены специальные обозначения: NG – именная группа, VG – глагольная группа, PredG – предложная группа, AdjG –группа прилагательного и др.)
- Лемма – все слова словосочетания в начальной форме (например, *подкладывать свинью – подкладывать свинья*). В случае однословных слов лемма совпадает с вокабулой.

- Тональность. Авторы выделяют три тональности: *positive* (положительная), *negative* (отрицательная), *neutral* (нейтральная). Некоторым словам приписан тэг *positive/negative* (например, словам *барьер, верноподданный, опьянеть*). Тональность обозначается одним из вышеперечисленных тэгов, написанных латиницей.
- Источник тональности. Авторы называют эту категорию источником, хотя речь скорее идёт о некоторых иллокутивных характеристиках слова. Эта категория также принимает три значения: *opinion* (мнение), *feeling* (чувство) и *fact* (факт).
- Некоторые вхождения сопровождаются комментариями. Это делается в тех случаях, когда тональность отличается для разных значений многозначного слова, и в связи с этим перечисляются все значения слова по тезаурусу РуТез (на основе которого создан словарь, см. [Лукашевич 2011]).

Вокабула	Часть речи	Лемма	Тональность	Источник	Примечания
яркий	Adj	яркий	neutral	opinion	"ЯРКИЙ ПО СВЕТУ"
яркий	Adj	яркий	neutral	opinion	"ЯРКИЙ ПО ЦВЕТУ"
яркий	Adj	яркий	positive	opinion	"ЯРКИЙ (ВЫРАЗИТЕЛЬНЫЙ)"

Таб. 13. Пример многозначного слова в словаре РуСентиЛекс

Следует отметить ряд особенностей словаря, существенных при его применении для конкретной задачи. Первое касается объёма словаря. Словарь содержит в общей сложности 16 058 вокабул, из которых 11 920 (74%) составляют единичные слова, а оставшиеся 26% приходятся на словосочетания от 2 до 6 слов. При этом 1743 вокабулам (чуть более 1%) приписана нейтральная тональность. Подход вызывает вопросы: если словарь содержит лишь тонально окрашенные слова, зачем включать нейтральные? Если словарь включает нейтральные слова наравне с тонально окрашенными, почему он не включает всех слов русского языка, которых, несомненно, значительно больше шестнадцати тысяч?

Второе замечание касается состава словаря (словника). Основу словаря составляет РуТез [Лукашевич 2011]. Казалось бы, словарь отражает состояние современного литературного русского языка. Однако в нём содержатся и такие, казалось бы, далёкие от устоявшихся норм слова, как «анфолловить», «анфоловить» (в обоих вариантах написания), «забанить» и некоторые другие. Получается, что, с одной стороны, словарь содержит не только кодифицированный литературный язык, но и некоторый жаргон, который, впрочем, представлен крайне фрагментарно.

Последнее замечание касается технической реализации словаря. Словарь распространяется в формате текстового файла, пригодного для обработки различными компьютерными программами. С сожалением приходится отметить некоторое количество опечаток, досадным образом нарушающих структуру данных или не дающих идентифицировать слова.

Несмотря на это, словарь представляет немалую академическую ценность. Тем не менее, в исходном виде словарь трудно применить для решения поставленной задачи. В рамках настоящего исследования был принят ряд решений, направленных на улучшение качества обработки имеющихся данных с помощью словаря RuSentiLex и на упрощение процедуры создания признакового пространства.

Во-первых, было необходимо очистить словарь от опечаток, чтобы исключить ошибочное нераспознавание каких-либо слов. Во-вторых, было принято решение очистить словарь от нейтральных слов, чтобы избежать внесённой разработчиками несбалансированности. В-третьих, было принято решение рассматривать только единичные слова (не словосочетания), имеющиеся в словаре. Такое решение кажется оправданным, поскольку словосочетаний в словаре не так много, следовательно, мы потеряем не очень много данных. Кроме того, оценка наличия последовательности незаданной длины в строке высказывания связано с алгоритмическими сложностями, которые кажутся неоправданными в контексте места данного этапа в общей картине исследования. В некоторой перспективе, однако, возможно развитие данного метода с привлечением большего объёма словарных данных и более сложных алгоритмов.

На основании данных, полученных из словаря, можно подсчитать отдельно количество положительно и отрицательно окрашенных слов в высказывании, количество слов от каждого «источника», а также долю эмоционально окрашенных слов в предложении, долю положительных слов среди всех эмоциональных и другие характеристики. Подробный список всех признаков можно найти в приложении. Всего было извлечено 38 признаков, что привело к созданию векторов небольшой размерности. Тем интереснее сравнить это пространство с пространством, полученным с помощью алгоритма doc2vec (имеющим размерность 100).

В результате, был применён алгоритм, состоящий из следующих шагов:

1. Алгоритм получает на вход реплику и создаёт для неё вектор размерностью 38. Изначально значение для каждого признака равно 1 (во избежание возможного деления на нуль применяется метод Лапласа, прибавляющий 1 к каждому значению).

2. Алгоритм лемматизирует каждое слово в реплике средствами библиотеки Pymorphy2 и ищет его в словаре (по леммам).
3. Если слово найдено в словаре, извлекаются две его характеристики: тональность и «источник».
4. Сначала подсчитывается количество тонально окрашенных слов в реплике. Значения всех признаков 1-14 (см. Приложение), в названии которых есть извлечённое значение тональности и «источника» увеличиваются на 10 (условное число, выбранное для простоты расчтов и значительно превышающее 1). В случае, если тональность имеет значение *positive*/*negative*, значение увеличивается на 5 (поскольку средств отличить положительный контекст от отрицательного на данном этапе исследования нет).
5. После того, как все слова в реплике обработаны и счёт абсолютных вхождений каждого из параметров завершён, рассчитываются доли слов, т.е. признаки 15–38 (см. Приложение).
6. Вектор для реплики добавляется в общую матрицу, и алгоритм переходит к следующей реплике.

Признаковое пространство (в дальнейшем – К-пространство), полученное этим методом, математически ничем не отличается от любого другого, и потому можно провести операции, подобные тем, что были проведены на материале пространства doc2vec.

Ниже представлены те же тестовые примеры, что и для А-пространства, и ближайшие к ним высказывания.

Пример 1. Высказывание со словами, отрицательно характеризующими собеседника.

Видно, что слово *глупый*, имеющееся в примере (предложение 0) повторяется только п предложении 1. В предложении 2 есть синонимичные ему существительные *чепуха*, *ерунда, чушь*. В предложении 3 есть несколько однокоренных слов отрицательной тональности с одинаковой семантикой, но не с семантикой глупости. В предложении 4 остаётся конструкция с повторением, но исчезает отрицательная тональность. В предложении 5 не повторяются ни слова, ни конструкции из предложения 0; единственное слово, которое имеет отрицательное значение – *озабот* – не включено в словарь.

0	ты дурак . я давно замечала что инфы - глупые . зачем моей сестре нужны были эти глупышки
1	ругаешься ? смутила я тебя , однако . а насчёт глупых твоих мыслишек можно так поразмыслить . итак
2	чепуха ! ерунда ! чушь собачья ! я уверена в этом ! и я могу это доказать !
3	не спорь со мной ! ты врунья ! и где только так врать научилась ! всё враньё !
4	я тебе ? а ты мне ? я тебе ? ну может быть
5	потому что не хочу в твоих глазах сразу выглядеть каким нибудь озаботом .

Таб. 14. Ближайшие синонимы Примера 1 в К-пространстве

Пример 2. Высказывание со словами, выражающими положительные эмоции.

Слова с положительной тональностью можно встретить в предложениях 1 и 2 (*добрый*), предложении 4 (*хорошее*) и предложении 5 (*интересная*). Также можно отметить некоторое сходство очень общего синтаксического строения всех высказываний, а именно сохранение двух пунктуационно выделенных частей.

0	это просто замечательно ! рад за тебя .
1	добрый день) располагайся , инф пеппи)
2	добрый день , мы же уже здоровались)
3	время никуда не идет , оно просто уходит
4	если хочешь услышать о себе хорошее - умри
5	жизнь - интересная штука ! всем привет !

Таб. 15. Ближайшие синонимы Примера 2 в К-пространстве

Пример 3. Высказывание, не содержащее эмоциональной лексики, употребляющееся в определённых ситуациях.

Как и следовало ожидать, результаты для высказываний, не содержащих эмоционально окрашенной лексики, не кажутся близкими.

0	здравствуйте ! представьтесь пожалуйста !
1	нормалек) а ты ?
2	что бы рассказать – то
3	любой вопрос можно считать философским
4	вот такой я молодец !
5	не спорь со старшими !

Таб. 16. Ближайшие синонимы Примера 3 в К-пространстве

Пример 4. Высказывание, не содержащее эмоционально окрашенной лексики и каких-либо конструкций.

В данном примере также можно проследить сходство общего характера: все найденные высказывания достаточно длинные и состоят из нескольких пунктуационно выделеных частей.

0	со мной все понятно , я читатель детективов ищущий носочки . а кто ты по профессии ?
1	ага , настроение - . . . . добавь в двух словах , как ты умеешь)
2	хорошо , что ты согласен . а хочешь , я у тебя приму экзамен по географии ?
3	и не говори . весь год работаешь , как лошадь , а приходит отпуск и пролетает незаметно
4	я думаю , что на бал прилично поехать на роликах . ты на роликах умеешь кататься ?
5	это пушкин написал . но вы это еще небось не проходили . ты учишься в школе ?

Таб. 17. Ближайшие синонимы Примера 4 в К-пространстве

В целом можно отметить, что алгоритм хорошо справляется с поиском близких высказываний к эмоционально окрашенным репликам, а также способен уловить некоторые общие характеристики высказываний.

Гибридный метод

Оба вышеописанных метода имеют ряд достоинств и недостатков. Представляется, что сочетание этих методов могло бы позволить избавиться от последних и совместить в рамках одного признакового пространства первые.

Как уже было показано, автоматический нейросетевой алгоритм хорошо справляется с задачей извлечения большого количества признаков. Однако эти признаки не вполне соответствуют тем, которые значимы в рамках настоящего исследования. Поэтому за основу гибридного метода был взят автоматический метод, но на определённом этапе результаты подверглись оценке человеком.

Ниже подробно описан представляемый алгоритм.

Первым шагом этого алгоритма является представление слов в виде векторов средствами алгоритма word2vec. Параметры модели подбираются вручную. Оценка результатов производилась так же, как и в других случаях. Таким образом, получаются достаточно точные векторные представления отдельных слов. Однако, как показали результаты автоматического метода, этого недостаточно, и необходимо большее обобщение.

Для этого была использована автоматическая кластеризация. Кластеризация подразумевает объединение слов, векторные представления которых близки, в группы. С лингвистической точки зрения, кластеризация – своеобразный способ поиска групп слов, имеющих сходное контекстное распределение. Для кластеризации на базе алгоритма word2vec в рамках пакета word2vec создан алгоритм word2clusters, наследующий его архитектуру. Число кластеров – такой же параметр модели, как размерность вектора или ширина окна – подбирается вручную. Однако подобрать параметры для модели алгоритма word2clusters оказывается значительно сложнее, чем для алгоритма word2vec. Идеальным был бы такой результат, при котором те высказывания, которые оказываются ближайшими при определении косинусной близости, попадают в один кластер, а более далёкие высказывания попадают в другие кластеры. Однако ситуация осложняется лингвистической реальностью. В действительности одно и то же слово может иметь несколько значений и быть близким к нескольким различным словам. Поэтому кластеры оказываются очень разными по объёму (от 0 до 209 слов[13]) и неравномерными по плотности. Разумеется, это оказывает значительное влияние на точность работы алгоритма.

Для преодоления этих трудностей был принят ряд мер. Во-первых, кластеры были вручную оценены как приоритетные и неприоритетные. Приоритетными были сочтены такие кластеры, которые содержали словоформы, с большой долей вероятности способные

[13] На предварительном этапе было получено 1500 кластеров, из которых 77 пустых.

оказаться в репликах, свидетельствующих об отрицательной оценке собеседника. Такими словами были признаны следующие:

- субстантивные отрицательно оценочные номинации людей (*дура, недоумок*),
- адъективные отрицательно оценочные характеристики людей (*глупый, тупой, сумасшедший*),
- глаголы рече- и мыслепорождения (*сказать, подумать*),
- глаголы рече- и мыслевосприятия (*услышать, понять*),
- глаголы, связанные с обучением (*учиться, развиваться*).

Ввиду отсутствия лемматизации и значительной вариативности приходилось вручную просматривать состав каждого кластера.

Приоритетными были признаны такие кластеры, в которых вышеперечисленные слова составляли не менее 5%. Остальные кластеры были сочтены неприоритетными.

Затем, для обобщения значения слов, была оценена близость вектора каждого слова к центру каждого кластера. Центры кластеров вычислялись при помощи библиотек Numpy и Scipy. Для оценки близости использовалась косинусная мера. Таким образом можно было выявить многозначные слова и оценить их более адекватно.

На этапе оценки близости слов к центрам кластеров использовалась информация о приоритетности кластеров. Так, косинусная мера близости для неприоритетных кластеров была уменьшена путём деления на некоторую константу. Константы подбирались вручную, оценка результатов производилась качественно. Наилучшие результаты показало деление на 3[14]. В результате этого шага был осуществлён переход от векторов размерности, использовавшейся для тренировки модели, к размерности, равной количеству непустых кластеров. Этот переход очень важен, поскольку именно он обеспечивает семантическое обобщение и отрыв от конкретной лексической информации, то есть, избавляется от главного недостатка автоматического метода. Физический смысл новых измерений заключается в том, что каждое из них кодирует, насколько слово близко к усреднённой семантике каждого кластера.

Векторы высказываний были получены усреднением векторов всех входящих в них слов. В итоге было получено признаковое пространство размерностью 1423 (далее Г-пространство). Ниже можно видеть результаты.

[14] Казалось бы, можно было бы увеличить значение косинусных мер для приоритетных кластеров, а не уменьшать их для неприоритетных для экономии машинных ресурсов (поскольку приоритетных кластеров меньше). Однако в таком случае необходима была бы дополнительная нормализация всех векторов, потому что умножение чисел от 0 до 1 на натуральное число привело бы к тому, что некоторые значения могли бы оказаться больше единицы. В то же время деление на натуральное число не приводит к такой необходимости.

Пример 1. Высказывание со словами, отрицательно характеризующими собеседника.

Как видно, в предложениях 1–5 не повторяются слова и конструкции, использованные в примере (предложении 0). Также не наблюдается повторяющихся синтаксических конструкций. Однако кажется, что во всех предложениях присутствует общая семантика удивления и недоумения, выраженная риторическим вопросом в предложении 0. В предложении 1 есть противительный союз *а*, выражающий несоответствие между ожиданием говорящего и действительностью. В предложении 2 также есть противительный союз *а*. В предложении 5 предложение с *вдруг* также привносит семантику удивления. В предложениях 3 и 4, впрочем, очевидных синтаксических и семантических маркеров такой семантики нет.

0	ты дурак . я давно замечала что инфы - глупые . зачем моей сестре нужны были эти глупышки
1	полгода назад , а ты сказала нельзя - родители не поймут
2	и не говори . весь год работаешь , как лошадь , а приходит отпуск и пролетает незаметно
3	правда - хорошее слово . раньше даже газета такая выходила . ее еще до революции выпускать стали . кстати , ты кого знаешь из революционеров ?
4	ну вот ещё одному делать не чего , как кликать бес толку
5	проверь своё чувство ритма . а вдруг оно есть ?

Таб. 18. Ближайшие синонимы Примера 1 в Г-пространстве

Пример 2. Высказывание со словами, выражающими положительные эмоции.

В этом примере несомненно сходство предложений по тому, что все они восклицательны. В предложениях 1-3 есть слова и конструкции, использующиеся для подтверждения слов собеседника, как и конструкция *это просто замечательно*. В предложениях 4-5 есть личные местоимения, как и в предложении 0.

0	это просто замечательно ! рад за тебя .
1	ага так и есть !
2	и я хочу такое же !)
3	ну , это хорошо ! у меня тоже всё отлично ! а как тебя зовут ?
4	что у меня их не м . б . !)
5	мда . . . ну если не знать как тебя зовут ! короче тебя зовут !

Таб. 19. Ближайшие синонимы Примера 2 в Г-пространстве

Пример 3. Высказывание, не содержащее эмоциональной лексики, употребляющееся в определённых ситуациях.

В данном примере также все ближайшие высказывания сохраняют восклицательность. Следует отметить, что глаголы в предложениях 1-2 стоят в императиве, как и в предложении 0. Можно также постулировать наличие семантики императивности в предложениях 3 (*ой, хватит = перестань*) и 4 (*надо закрыть = закрой*).

0	здравствуйте ! представьтесь пожалуйста !
1	не жди, пока ударят! дай мобилу!
2	да! купи мне десять пачек!
3	ой блин хватит яподстолом валяюсь ! ! ! !
4	быстрей надо закрыть ! ! !
5	а ! пожиратель смерти ! ! !

Таб. 20. Ближайшие синонимы Примера 3 в Г-пространстве

Пример 4. Высказывание, не содержащее эмоционально окрашенной лексики и каких-либо конструкций.

Во всех предложениях, кроме 4, есть вопросительность, как и в предложении 0. Также во всех предложениях, кроме 5, есть обособленные запятыми части. Семантические и дискурсивные повторения проследить не удаётся.

0	со мной все понятно , я читатель детективов ищущий носочки . а кто ты по профессии ?
1	что за намеки , лапа ?
2	по ночам ничего в окна не скреблось , не завывало ?
3	я что , по - вашему , заводной апельсин ?
4	мне в трамвае счастливый билет дали , я загадал что ты поговоришь со мной , как видишь сбылось !
5	ты никогда не думал о карьере торгаша шаурмой ? у тебя просто дар говорить о кошатине и собачатине

Таб. 21. Ближайшие синонимы Примера 4 в Г-пространстве

Как можно убедиться, между примерами и ближайшими к ним высказываниями, найденными в Г-пространстве, есть связи на уровне очень обобщённой семантики, интонации и прагматики.

Результаты применения всех трёх признаковых пространств различны и интересны, поэтому было принято решение использовать в дальнейшем все три набора признаков.

Раздел 4: Классификация

Постановка проблемы

Несмотря на значительный объём проделанной работы и ряд интересных наблюдений, создание признакового пространства – лишь предварительный этап для достижения поставленной цели в рамках настоящего исследования, т.е. автоматизированного извлечения реплик, свидетельствующих об отрицательной оценке собеседника, из диалогов.

Для успешного достижения поставленной цели необходимо правильно выбрать методы. Традиционно для решения задач извлечения чего-либо (например, извлечения коллокаций или именованных сущностей) решается с опорой на различные словарные методы. В случае работы с высказываниями такие методы оказываются неприменимыми. Вышеописанные методы работают на уровне слов, а для достижения нашей цели необходимо работать на уровне предложения.

Цель исследования можно переформулировать. Извлечение реплик из коллекции реплик можно понять как задачу отделения одних реплик от других. Искомые реплики и остальные реплики – элементы одного уровня (в отличие, например, от именованных сущностей, извлекаемых не из коллекции других сущностей, а из единиц другого типа). Таким образом, отделение одних реплик от других – задача бинарной классификации. Каждая реплика является элементом, объектом рассмотрения. Весь корпус диалогов – коллекция элементов. Реплики, свидетельствующие об отрицательной оценке собеседника, – элементы первого класса. Остальные реплики – элементы другого класса. Задача сводится к максимально точному отделению реплик одного класса от реплик другого класса.

Однако решение этой задачи даже в форме задачи бинарной классификации сопряжено с рядом сложностей. Во-первых, постановка задачи как задачи бинарной классификации, основывается на гипотезе о том, что искомые реплики составляют единый класс. Эта гипотеза, однако, сомнительна. Для того, чтобы реплики составляли единый класс, необходимо, чтобы они обладали достаточным сходством по ряду признаков. Следовательно, успех классификации напрямую обусловлен признаковым пространством, на основе которого производится классификация. Именно поэтому были использованы все три признаковых пространства, полученные на предыдущем этапе исследования, несмотря на недостатки каждого из них.

Вторая сложность сопряжена с несбалансированностью имеющихся данных. Реплик, свидетельствующих об отрицательной оценке собеседника, в любом корпусе заведомо меньше половины. (Действительно, если бы было так, получалось бы, что один собеседник постоянно недоволен другим. Если такие диалоги и происходят между людьми

и чат-ботами, они, как правило, длятся недолго.) Точную долю искомых реплик в имеющемся корпусе предсказать невозможно, поскольку она зависит от большого числа параметров: особенности состава корпуса, обученность отдельных ботов, индивидуальные речевые особенности собеседников. Получается, что задача сводится к разделению двух классов неизвестного объёма. Это противоречит традиционной постановке задачи классификации, при которой классы представлены в коллекции данных поровну. Проблема несбалансированности данных для классификации возникает и в других областях, самой распространённой из которых является, пожалуй, диагностика различных заболеваний (см., в частности, [Mena & Gonzalez 2008]). Различные методы работы с несбалансированными данными рассматриваются ниже подробнее.

Третья сложность возникает в результате того, что традиционно методы классификации применяются к размеченным данным. Для обучения классификаторов используются объёмные обучающие выборки, где класс, к которому необходимо отнести класс, уже заранее известен. Например, в качестве таких баз данных могут быть использованы показатели анализов пациентов, у которых уже было констатировано наличие или отсутствие болезни. Благодаря такой процедуре обучения, классификатор «выучивает» характерные признаки единиц каждого класса, что позволяет ему впоследствии предсказать класс для каждого нового элемента.

Однако настоящее исследование не имеет прецедентного корпуса. И дело не столько в том, что до сих пор не было собранно коллекции человеко-машинных диалогов на русском языке, сколько в разности поставленных задач. Задача автоматизированного извлечения каких-либо реплик из какого-либо корпуса никак не может иметь прецедентного корпуса в принципе, поскольку может быть решена ровно один раз. Если корпус будет пополняться или если метод будет распространён на другие коллекции данных (например, подключён к диалоговой системе как модуль синхронной проверки системы на успешность), можно будет каждую новую входящую реплику и принимать решение, относится ли она к искомым репликам. (Подобно тому, как данные каждого нового пациента оцениваются отдельно от других, и только на их основе принимается решение). Тем не менее, этап собственно деления корпуса на две части должен основываться сразу на всех данных корпуса без ручной разметки (поскольку ручная разметка делает ненужной автоматическую).

В различных исследованиях, применяющих и оценивающих методы классификации, предлагаются различные стратегии, которые подробнее рассматриваются в следующем разделе.

Обзор существующих решений

В случае несбалансированности данных можно прибегнуть к стратегиям балансировки выборки, то есть выбрать равное количество примеров из обоих классов, несмотря на то, что они будут представлять разные доли от своих классов. Таким образом можно добиться лучших результатов для определённых классификаторов (подробнее см., в частности, в [Estabrooks, Jo, Japkowicz 2011; Laurikkala 2001; Weiss & Provost 2001].

В [He & Garcia 2009] описано семь различных алгоритмов модификации выборки, один из которых также включает в себя модификацию процесса обучения (бустинг). Мы не будем подробно останавливаться на каждом из них, однако отметим, что разные алгоритмы демонстрируют максимальную результативность при обучении на различных типах данных.

В статье [Mena & Gonzalez 2008] отмечается, что искусственное увеличение количества примеров из меньшего класса может привести к переобучению. Тем не менее, авторы выступают скорее за этот метод, указывая на то, что малая представленность какого-либо класса может быть следствием недостаточного размера корпуса и, следовательно, модификация выборки поможет исправить недостатки сбора данных. В нашей задаче, несмотря на относительно небольшой объём данных, несбалансированность классов продиктована естественными причинами, а не особенностью сбора данных.

Создание специфических выборок может оказаться полезным для решения задачи, поставленной в настоящем исследовании, в том числе потому, что потребует минимальных усилий для предварительной разметки. Выборки могут быть созданы различными способами, из которых чаще всего применяется случайный. В последнее время набирает популярность метод активного обучения (active learning), при котором одновременно производится предварительное деление данных на части и ручная разметка. Активное обучение в последнее время всё чаще применяется для задач классификации. Этот метод выгодно отличается от других тем, что не требует разметки всех данных. В [He & Garcia 2009] описывается алгоритм активного обучения на основе метода опорных векторов (SVM), при котором выбираются наиболее информативные примеры, лежащие вблизи разделяющей гиперплоскости с тем, чтобы скорректировать её положение. Однако метод опорных векторов следует применять с осторожностью на небольшом корпусе несбалансированных данных во избежание переобучения.

В статье [Lughofer 2011] продемонстрирован алгоритм активного обучения для многоклассовой классификации. Он подразумевает два этапа. Сперва все примеры кластеризуются, чтобы облегчить последующую процедуру классификации. Второй этап предполагает выборку и разметку примеров из каждого кластера. Из каждого кластера

выбираются наиболее центральные примеры, чтобы по ним можно было ориентироваться. После этого выбираются примеры, находящиеся на границах между кластерами, чтобы скорректировать деление на кластеры. Однако такой метод нацелен, во-первых, на многоклассовую классификацию, а не бинарную, а во-вторых, на сбалансированные данные. Тем не менее, этот алгоритм оказался действенным при подготовке данных для обучения (см. далее).

Поправка на несбалансированность корпуса, равно как и на отсутствие в нём разметки, может производиться на этапе выбора алгоритма. Наиболее часто используемые алгоритмы классификации, например, Наивный Байесовский классификатор или метод К ближайших соседей в чистом виде неприменимы к таким данным, поскольку очень чувствительны к несбалансированности ([Sun 2007]).

Перспективными представляются методы одноклассовой классификации (one-class classification), получившие распространение в последнее время. Одноклассовая классификация так же, как и бинарная, делит весь корпус на две части. Одноклассовой она называется потому, что учится предсказывать только один класс. Иными словами, при предъявлении каждого нового объекта перед объектом бинарной классификации ставится вопрос «на объекты какого из двух классов этот объект похож больше?», а перед алгоритмом одноклассовой классификации ставится вопрос «достаточно ли объект похож на объект единственного искомого класса?». Такие алгоритмы специально созданы для работы с несбалансированными данными и хорошо с ними справляются. В [Barnabé-Lortie 2015] было доказано, что чем более несбалансированным является набор данных, тем больше одноклассовая классификация превосходит двухклассовую.

Как правило, при работе с несбалансированными данными объекты миноритарного класса (то есть те объекты, которых в корпусе меньше) считаются целевыми объектами, а остальные объекты считаются шумом. При этом тренировка может включать как объекты только одного класса (авторы [Mena & Gonzalez 2008] сравнивают такой подход с экстремальной модификацией выборки в пользу меньшего класса), так и объекты обоих классов.

Кроме того, такие методы легко учатся на небольшом количестве примеров, поскольку рассчитаны на применение в тех случаях, когда данных искомого класса очень мало (безотносительно общего объёма корпуса).

Одноклассовая классификация кажется не только удобным с точки зрения технической реализации подходом к решению задачи, поставленной в нашем исследовании, но и методологически верным. Действительно, мы предполагаем некоторую гомогенность класса реплик, свидетельствующих об отрицательной оценке пользователя, и поэтому

стараемся их выделить. В то же время ошибочно было бы предполагать гомогенность тех реплик, которые останутся в корпусе после извлечения реплик, свидетельствующих об отрицательной оценке пользователя. Они будут слишком разнородными, чтобы представлять собой самостоятельный класс. В связи с этим задача бинарной классификации не имеет физического смысла, поскольку двух классов, каждый из которых обладает некоторым особым набором характеристик, не существует. Задача поиска одного класса, напротив, весьма осмысленна.

В итоге для исследования было выбрано пять методов классификации, более или менее пригодных для оценки несбалансированных неразмеченных данных. Подробнее об особенностях из архитектуры и специфике применения к нашей задаче см. ниже.

Создание и обучение моделей

Подготовка выборки

После того, как были выбраны и запрограммированы методы классификации, стало необходимо проверить, насколько успешно они работают с имеющимися данными, а затем применить наиболее успешные модели ко всему корпусу, чтобы извлечь искомые реплики.

Для начала этапа обучения было необходимо создать выборки. Поскольку все методы способны работать на небольшом количестве примеров, было принято решение создать две выборки по 500 реплик в каждой. Выборки создавались на основе алгоритма [Lughofer 2011]

Алгоритм предполагает автоматическую кластеризацию результатов на первом этапе и затем последовательное уточнение состава каждого кластера. Однако поскольку этот алгоритм рассчитан на многоклассовую классификацию сбалансированных данных, его применение к нашей задаче не вполне оправдано. Однако с помощью первого этапа алгоритма удалось получить необходимые выборки:

- Обучающую выборку искомого класса (ОИ): 500 реплик, содержащих реплики, свидетельствующие об отрицательной оценке собеседника
- Обучающую выборку фонового класса (ОФ): 500 реплик, не содержащих реплики, свидетельствующие об отрицательной оценке собеседника
- Тестовую выборку: 100 реплик, 20 из которых относятся к искомому классу и 80 – к фоновому.

Все отобранные реплики были нами просмотрены и их принадлежность к тому или иному классу подтверждена.

Выбранные методы

Autoencoder

Одно из возможных решений бинарной классификации методами обучения без учителя было представлено в статье [Japkowicz 2001]. Автор предлагает тренировать нейронную сеть на примерах только одного класса (искомого)[15]. Тогда, по мнению авторов, сеть будет хорошо справляться с примерами того же класса и плохо – с примерами другого класса. Строго говоря, такое обучение нельзя назвать обучением без разметки, поскольку для того, чтобы обучить сеть на примерах только одного класса их необходимо сначала найти.

Однако результаты, приводимые в статье, кажутся обнадёживающими. Обучение проводилось на очень маленьких корпусах (3 набора по 100 примеров размерности 256, 50, 60), что позволяет сделать предварительную частичную разметку небольшими усилиями. Использование только одного класса в качестве тренировочного позволяет решить проблему несбалансированности, поскольку в этом методе важны внутренние характеристики каждого примера, а не его отношение к другим примерам в выборке. То есть алгоритм извлекает некий инвариант, которым характеризуются примеры искомого класса, и оценивает допустимость вариативности. Получается, что если новый предъявляемый объект похож на инвариант в достаточной мере, не имеет значения, на что он похож ещё. Такой подход устойчив не только к вышеназванным проблемам несбалансированности и неразмеченности, но также и к сложности данных самих по себе, что весьма удобно для целей настоящего исследования.

За основу был взят алгоритм, описанный в статье [Japkowicz 2001] и модифицирован для работы с имеющимися данными средствами библиотеки Keras для языка Python.

Автоэнкодер получает на вход вектор диалоговой реплики. Затем сворачивает его в вектор заданной **размерности скрытого слоя**. На основании рекомендаций документации к библиотеке Keras, а также ряда пробных моделей было принято решение принять размерность скрытого слоя равной 10% размерности вектора, а именно

- равной 4 для признакового пространства размерностью 38,
- равной 10 для признакового пространства размерностью 100,
- равной 140 для признакового пространства размерностью 1423.

Внутренним критерием успешности алгоритма считается минимальная ошибка. На каждом шаге алгоритм стремится натренироваться так, чтобы ошибка была меньше, чем на

[15] В подобном духе реализован алгоритм, описанный в [Mena & Gonzalez 2008], только в нём для обучения используются оба класса. По данным авторов, он превосходит подобные алгоритмы, созданные ранее, однако в рамках нашей задачи обучение на одном классе предпочтительнее.

предыдущем. **Тип ошибки**, которая будет минимизироваться, можно настроить. Нами была выбрана косинусная близость, поскольку эта метрика уже не раз использовалась в настоящем исследовании для оценки результатов. Для всех моделей в конце тренировки была получена ошибка меньше -0.8.

С типом ошибки тесно связан **тип оптимизатора** обучения. Не вдаваясь в математические и технические подробности обзора возможных вариантов, отметим, что наилучшие результаты за наименьшее время были получены оптимизатором ‘nadam’.

Количество шагов тренировки каждой модели подбиралось таким образом, чтобы обучение произошло до конца, т.е. ошибка достигла минимально возможного для имеющихся данных значения.

Остальные параметры не были модифицированы.

В результате получилось, что сбалансированная F-мера (F_0) не превышает 0,6. Самый высокий показатель достигнут в А-пространстве. Для всех пространств F-мера с приоритетом точности несколько превышает F-меру с приоритетом полноты. Это значит, что метод достаточно точно определяет искомые реплики, однако делает это для лишь небольшой доли всех необходимых реплик.

В целом, результаты работы нейронной сети оказались весьма средними.

One-Class Support Vector Machine

Метод опорных векторов часто применяется для задач классификации. Идея метода заключается в том, что алгоритм стремится наиболее точно провести гиперплоскость, разделяющую два класса по всем измерениям. В случае одноклассовой классификации, алгоритм стремится на основании данных одного класса создать такую гиперплоскость, по одну сторону которой могли оказаться другие элементы того же класса, а по другую сторону – элементы другого класса.

Алгоритм OCSVM реализован в рамках пакета SciKit-Learn ([Schölkopf et al. 2001]). Настройка алгоритма позволяет модифицировать ряд параметров.

Тип ядра модели зависит от особенности распределения данных. Для признакового пространства малой размерности (К-пространства) лучшие результаты были получены на основе нелинейного ядра(‘rbf’). Для признаковых пространств большей размерности (А-пространства и Г-пространства) лучше оказалось линейное ядро

Для нелинейного ядра также можно задать **гамма-коэффициент**, по умолчанию равный 1/размерность. Для всех признаковых пространств были созданы модели с различными коэффициентами, что способствовало улучшению результатов для всех моделей. Лучшими оказались модель с коэффициентом 0.1 для К-пространства и 0.2 для Г-

пространства. Для А-пространства модель с нелинейным ядром при всех протестированных значениях параметра показывала худшие результаты, чем модель с линейным ядром.

Также можно настраивать **ню-параметр**, кодирующий максимальное количество допустимых ошибок при тренировке и минимальную долю опорных векторов. Тонкая настройка этого параметра помогла весьма существенно улучшить результаты для всех трёх пространств. В итоге лучшими были признаны модели со следующими значениями:

- К-пространство: 0.13
- А-пространство: 0.25
- Г-пространство: 0.15

Во всех трёх признаковых пространствах были показаны результаты, при которых сбалансированная F-мера превышает 0,5. В А-пространстве получена очень высокая полнота – 83%. В Г-пространстве сразу несколько моделей демонстрируют F-меру с приоритетом точности, близкую к 0,6 и превышающую её. Эти результаты говорят о том, что метод OCSVM особенно хорошо применим в пространствах большой размерности. Однако и в К-пространстве одна модель продемонстрировала весьма высокую полноту (0,71) и демонстрируют F-меру с приоритетом точности (0,625).

Метод OCSVM продемонстрировал достаточно хорошую точность при обучении на всех трёх признаковых пространствах. Эффекта переобучения (что является самым распространённым недостатком этого метода) не наблюдается.

Elliptic Envelope

Этот алгоритм основан на изучении данных путём сравнения расстояния Махаланобиса между ними ([Rousseeuw & Van Driessen 1999]). Изначально предполагалось, что такой метод применим только к нормально распределённым данным, однако ряд работ подтвердил, что это не так. В целом, метод показал себя успешно справляющимся с несбалансированными данными, что позволило ему быть применённым к нашей задаче.

Единственным параметром, который был модифицирован в рамках работы с этим методом, была **контаминация**. Под контаминацией понимается количество выделяющихся объектов в выборке. Обучение алгоритма производилось на обоих классов и наилучшие результаты были достигнуты при соотношении примеров искомого и фонового класса в обеих выборках как 2:8. Однако не известно, окажется ли алгоритм столь же результативным на всём корпусе, потому что доля искомых реплик в нём не известна.

Тем не менее, результаты тестов позволяют заметить высокое качество моделей Elliptic Envelope в А-пространстве и Г-пространстве. В А-пространстве одна из моделей показала F-меру с приоритетом точности равную 0,75, что позволяет говорить об успешных

и даже конкурентноспособных результатах. В Г-пространстве одна из моделей показала рекордно высокую точность 0,9 (правда, при полноте 0,5). В К-пространстве метод продемонстрировал достаточно низкие результаты, что, по-видимому, объясняется недостатком размерности.

Local Outlier Factor

Метод поиска локальных выбросов основан на оценке ближайшего окружения каждого объекта. С точки зрения общей логики алгоритма он похож на один из самых популярных методов кластеризации – k ближайших соседей, который плохо применим к задачам бинарной классификации. Метод поиска локальных выбросов оценивает плотность распределения соседей вокруг каждого объекта и сравнивает её с плотностью распределения соседей вокруг каждого из соседей. Если наблюдается существенное различие, точка определяется как выброс ([Breunig et al. 2000]).

Удобство этого метода для настоящего исследования заключается в том, что модели могут быть обучены как на двух классах, так и только на одном. При обучении на двух классах модель уже на этапе обучения видит некоторые выбросы, и оценивает их соответственно. При обучении только на одном классе (фоновом) модель узнаёт необходимые сведения об объектах этого класса и новые объекты другого класса определяются как выбросы.

В рамках этого метода можно настроить **количество соседей**, на основании которых принимается решение, и **контаминацию** (см. выше). Для всех признаковых пространств лучшие результаты показали модели с учётом 30 соседей с уровнем контаминации 0.25. Обучение на одном классе показало несколько более низкие результаты.

Лучшие результаты были показаны в Г-пространстве. Одна из моделей продемонстрировала очень высокую точность (0,8) и весьма высокую сбалансированную F-меру (0,695). Также неплохие результаты были показаны в А-пространстве (полнота 0,8, правда, остальные показатели весьма средние). Результаты для К-пространства низкие.

Isolation Forest

Традиционно древесные методы применяются в задачах кластеризации, однако уже сложилась некоторая традиция их использования и для классификации. Метод изолирующего леса представляет собой попытку отделить какой-либо объект от всех остальных на основании деления всех объектов на две части последовательно по каждому из признаков ([Liu et al. 2008]).

Обучение этого метода основано на примерах, принадлежащих к обоим классам. Несмотря на то, что алгоритм, реализованный в рамках пакета Scikit Learn позволяет настраивать большое количество параметров, на наши результаты оказал существенное влияние лишь один – **контаминация**. Наилучшим оказался коэффициент 0.25. Однако, как уже отмечалось, неизвестно, насколько хорошими окажутся такие модели в применении ко всему корпусу.

В общем, алгоритм Isolation Forest показал весьма неплохие результаты только в А-пространстве, обеспечив полноту 0,8 и сбалансированную F-меру 0,76. Относительно неплохие результаты также показала одна модель в Г-пространстве (F_0 = 0,57).

Сводная таблица методов

В приведённой ниже таблице можно видеть метрики лучших моделей (с описанными выше модификациями параметров) для каждого метода. Модели, применимые к нескольким признаковым пространствам, даны несколько раз.

Со всеми моделями, параметрами и метриками можно ознакомиться в Приложении.

Метод	Пространство	Точность	Полнота	F-мера
Autoencoder	А	0,5	0,714286	0,588235
	Г	0,5	0,555556	0,526316
	К	0,5	0,555556	0,526316
OCSVM	А	0,5	0,833333	0,625
	Г	0,6	0,75	0,666667
	К	0,5	0,714286	0,588235
EE	А	0,6	0,857143	0,705882
	Г	0,9	0,5	0,642857
LOF	А	0,4	0,8	0,533333
	Г	0,8	0,615385	0,695652
IF	А	0,8	0,727273	0,761905
	Г	0,9	0,5	0,642857

Таб. 22. Результаты лучших моделей

Как видно, применение моделей к К-пространству оправдано только в случае двух методов. Также видно, что некоторые модели показывают более высокие результаты по параметру точности, а другие – по параметру полноты. Решение о том, какая метрика важнее, дискуссионно, именно поэтому были также посчитаны F-меры с приоритетом точности и с приоритетом полноты (см. Приложение). Тем не менее, при учёте моделей с равными или близкими показателями, мы отдавали предпочтение тем, у которых результат по полноте был выше. Этот приоритет объясняется постановкой задачи. Поскольку целью исследования является извлечение некоторых реплик, то кажется логичным поощрять те

модели, которые хорошо справляются с поиском таких реплик и достаточно полно их находят. Тот факт, что такие модели делают это не совсем точно, то есть, помимо искомых реплик извлекают ещё и некоторые другие, можно впоследствии компенсировать более тщательным анализом результатов извлечения. Более того, оценка точности не всегда проста, поскольку даже люди могут сомневаться в том, к какому классу реплик относятся некоторые экземпляры. Именно поэтому кажется более важным извлечь как можно больше реплик, похожих на искомые, а затем уточнить результат, если необходимо.

Все вышеперечисленные модели имеют право быть применёнными ко всему корпусу для достижения цели настоящего исследования.

Применение лучших моделей ко всему корпусу

Применение моделей ко всему корпусу сопряжено с проблемой оценки полученных результатов. Как видно из предыдущего раздела, разные модели всё же имеют разную результативность и совершают разные ошибки. Тем не менее, задача извлечения некоторых реплик из корпуса предполагает однозначный ответ – единственный набор извлечённых реплик.

Чаще всего говорят о трёх методах решения проблемы выбора результатов при наличии нескольких моделей:

1. Выбрать одну лучшую модель и использовать её результат (the-winner-takes-it-all approach)
2. Взять результаты нескольких моделей и суммировать результаты каждого типа для каждого объекта: каких больше, тот класс и приписать (voting approach)
3. Взять результаты нескольких моделей и суммировать с учётом надёжности модели (weighted voting approach)

Поскольку мы имеем несколько моделей не только с разным уровнем надёжности, но и с разной архитектурой и предрасположенностью к разным ошибкам (что было замечено в ходе лингвистического анализа результатов применения отдельных моделей), кажется разумным использовать эти знания для принятия конечного решения. Именно поэтому третий метод кажется единственно оправданным.

Тем не менее, остаётся дискуссионным вопрос о том, какой именно параметр следует учитывать для того, чтобы отделить более надёжные модели от менее надёжных. Мы предлагаем использовать для этого несколько параметров, и оценить, какой из них лучше подходит путём сравнения массивов реплик, извлечённых из корпуса с каждой с комбинацией моделей.

Было выбрано три метрики: сбалансированная F-мера, F-мера с приоритетом точности, F-мера с приоритетом полноты, средняя точность предсказания, точность предсказания класса искомых реплик.

Лучшие модели в каждом из пространств были применены ко всему корпусу. Каждая из моделей на выходе выдаёт решение об отнесении каждой из реплик к тому или иному классу. Затем значение каждой реплики (1 или -1) было умножено на соответствующий коэффициент (F-меру). Все отрицательные и все положительные результаты были суммированы. Конечное решение принималось на основе того, модуль каких результатов (положительных или отрицательных) был больше.

Следует сказать, что при применении к большому корпусу многие модели сбивались с того, чему они были обучены на материале небольших выборок и демонстрировали более низкие результаты. Так, модели, тестируемые в А-пространстве признаков, начали демонстрировать эффект переобучения и приписывать искомый класс едва ли не всем объектам. Модели, тестируемые в Г-пространстве, продемонстрировали такие результаты, что согласие между моделями оказалось невелико. Необходимость каким-то образом избавиться от этих эффектов ещё раз подтверждает правильность выбранного подхода для принятия конечного решения.

При сочетании результатов от всех вышеописанных моделей было получено 66535 реплик, извлечённых из корпуса. То есть, по мнению алгоритма, искомые реплики составляют около 23% корпуса, что, несомненно, является несколько завышенной оценкой. По всей видимости, даже метод сочетания и взвешивания моделей не помог полностью преодолеть проблему переобучения.

Раздел 5: Выводы

Из настоящего исследования можно сделать ряд выводов как лингвистического, так и методологического характера, которые помогут в дальнейшем развиваться направлению автоматической оценки диалоговых систем.

1. Реплики, свидетельствующие об отрицательной оценке собеседника, в русском языке могут оформляться различными лексическими и синтаксическими средствами, что делает их весьма сложными для изучения.
2. Выбранные методы классификации оправдывают своё применение в настоящем исследовании.
3. Значительная несбалансированность данных может быть частично преодолена на этапе обучения и тестирования, однако приводит к снижению результатов при применении к большему объёму данных.

4. Тем не менее, некоторые результаты обучения приближаются к получаемым в других исследованиях (в частности, [Gamon 2004]).
5. Признакового пространства малой размерности оказывается недостаточно для того, чтобы получить хорошие результаты классификации.
6. Признаковые пространства большой размерности демонстрируют разные результаты как на этапе тестирования, так и на этапе извлечения реплик, однако эти результаты могут быть высокими.

Цель настоящего исследования достигнута, но остаётся ряд интересных проблем, которые следовало бы решить в дальнейшем. Как нам кажется, следует проводить больше исследований на русскоязычном материале, поскольку это имеет как научную, так и практическую ценность. Создание корпуса человеко-машинных диалогов могло бы стать самостоятельным проектом. Улучшение качества уже имеющихся алгоритмов, а также поиск новых признаковых пространств могли бы позволить создать мощный инструмент оценки содержания диалоговых корпусов. Расширение материала исследования (например, извлечение также реплик с положительной тональности) могло бы позволить точнее оценивать диалоговые системы. Хочется надеяться, что в скором времени подобные исследования всё же приведут к созданию единому автоматизированному алгоритму оценки диалоговых систем.

Приложение 1

Номер признака	Что обозначает	Как считается
0	Количество положительно окрашенных слов	+10 за каждое слово, отмеченное в словаре как 'positive'
1	Количество негативно окрашенных слов	+10 за каждое слово, отмеченное в словаре как 'negative'
2	Количество спорно эмоционально окрашенных слов	+5 за каждое слово, отмеченное в словаре как 'positive/negative'
3	Количество слов, обозначающих факт	+10 за каждое слово, отмеченное в словаре как 'fact'
4	Количество слов, обозначающих чувство	+10 за каждое слово, отмеченное в словаре как 'feeling'
5	Количество слов, обозначающих мнение	+10 за каждое слово, отмеченное в словаре как 'opinion'
6	Количество положительно окрашенных слов, обозначающих факт	+10 за каждое слово, отмеченное в словаре как имеющее оба признака'positive' и 'fact'
7	Количество отрицательно окрашенных слов, обозначающих факт	+10 за каждое слово, отмеченное в словаре как имеющее оба признака'negative' и 'fact'
8	Количество спорно эмоционально окрашенных слов, обозначающих факт	+5 за каждое слово, отмеченное в словаре как имеющее оба признака'positive/negative' и 'fact'
9	Количество положительно окрашенных слов, обозначающих чувство	+10 за каждое слово, отмеченное в словаре как имеющее оба признака'positive' и 'feeling'
10	Количество отрицательно окрашенных слов, обозначающих чувство	+10 за каждое слово, отмеченное в словаре как имеющее оба признака'negative' и 'feeling'
11	Количество спорно эмоционально окрашенных слов, обозначающих чувство	+5 за каждое слово, отмеченное в словаре как имеющее оба признака'positive/negative' и 'feeling'
12	Количество положительно окрашенных слов, обозначающих мнение	+10 за каждое слово, отмеченное в словаре как имеющее оба признака'positive' и 'opinion'
13	Количество отрицательно окрашенных слов, обозначающих мнение	+10 за каждое слово, отмеченное в словаре как имеющее оба признака'negative' и 'opinion'
14	Количество спорно эмоционально окрашенных слов, обозначающих мнение	+5 за каждое слово, отмеченное в словаре как имеющее оба признака'positive/negative' и 'opinion'
15	Количество эмоционально окрашенных слов в высказывании	Сумма признаков 0, 1 и 2
16	Доля эмоционально окрашенных слов в высказывании	Количество эмоционально окрашенных слов / количество слов
17	Количество положительно окрашенных слов в высказывании	Сумма признаков 0 и 2
18	Доля положительно окрашенных слов в высказывании	Количество положительно окрашенных слов / количество слов
19	Доля положительно окрашенных слов среди всех эмоционально окрашенных	Количество положительно окрашенных слов / количество эмоционально окрашенных слов

20	Количество отрицательно окрашенных слов в высказывании	Сумма признаков 1 и 2
21	Доля отрицательно окрашенных слов в высказывании	Количество отрицательно окрашенных слов / количество слов
22	Доля отрицательно окрашенных слов среди всех эмоционально окрашенных	Количество отрицательно окрашенных слов / количество эмоционально окрашенных слов
23	Количество слов с субъективным источником тональности	Сумма признаков 4 и 5
24	Доля слов с субъективным источником тональности	Количество слов с субъективным источником тональности / количество слов
25	Доля слов с субъективным источником тональности среди всех эмоционально окрашенных	Количество слов с субъективным источником тональности / количество эмоционально окрашенных слов
26	Количество (возможно) отрицательно окрашенных слов, обозначающих факт	Сумма признаков 7 и 8
27	Доля (возможно) отрицательно окрашенных слов, обозначающих факт, в высказывании	Количество (возможно) отрицательно окрашенных слов, обозначающих факт / количество слов
28	Доля (возможно) отрицательно окрашенных слов, обозначающих факт, среди всех слов, обозначающих факт,	Количество (возможно) отрицательно окрашенных слов, обозначающих факт / количество слов, обозначающих факт
29	Доля (возможно) отрицательно окрашенных слов, обозначающих факт, среди всех эмоционально окрашенных	Количество (возможно) отрицательно окрашенных слов, обозначающих факт, / количество эмоционально окрашенных слов
30	Количество (возможно) отрицательно окрашенных слов, обозначающих чувство	Сумма признаков 10 и 11
31	Доля (возможно) отрицательно окрашенных слов, обозначающих чувство, в высказывании	Количество (возможно) отрицательно окрашенных слов, обозначающих чувство / количество слов
32	Доля (возможно) отрицательно окрашенных слов, обозначающих чувство, среди всех слов, обозначающих чувство	Количество (возможно) отрицательно окрашенных слов, обозначающих чувство / количество слов, обозначающих чувство
33	Доля (возможно) отрицательно окрашенных слов, обозначающих чувство, среди всех эмоционально окрашенных	Количество (возможно) отрицательно окрашенных слов, обозначающих чувство, / количество эмоционально окрашенных слов
34	Количество (возможно) отрицательно окрашенных слов, обозначающих мнение	Сумма признаков 13 и 14
35	Доля (возможно) отрицательно окрашенных слов, обозначающих мнение, в высказывании	Количество (возможно) отрицательно окрашенных слов, обозначающих мнение / количество слов
36	Доля (возможно) отрицательно окрашенных слов, обозначающих мнение, среди всех слов, обозначающих мнение	Количество (возможно) отрицательно окрашенных слов, обозначающих мнение / количество слов, обозначающих мнение
37	Доля (возможно) отрицательно окрашенных слов, обозначающих чувство, среди всех эмоционально окрашенных	Количество (возможно) отрицательно окрашенных слов, обозначающих чувство, / количество эмоционально окрашенных слов

Приложение 2

	Автоматический				
	precision	recall	F0	F(prec)	F(rec)
Autoencoder	0,5	0,714286	0,588235	0,625	0,555556
Elliptic Envelope contam=0.01	na	na	na	na	na
Elliptic Envelope contam=0.21	na	na	na	na	na
Elliptic Envelope contam=0.5	na	na	na	na	na
Elliptic Envelope default	0,6	0,857143	0,705882	0,75	0,666667
Isolation Frest contamination=0.01	0,1	1	0,181818	0,25	0,142857
Isolation Frest contamination=0.2	0,2	0,5	0,285714	0,333333	0,25
Isolation Frest contamination=0.25	0,8	0,727273	0,761905	0,75	0,774194
Isolation Frest contamination=0.3	0,7	0,777778	0,736842	0,75	0,724138
Isolation Frest contamination=0.4	na	na	na	na	na
Isolation Frest contamination=0.5	0,7	0,636364	0,666667	0,65625	0,677419
Isolation Forest default	0,5	0,714286	0,588235	0,625	0,555556
Local Outlier Factor default	0,4	0,666667	0,5	0,545455	0,461538
Local Outlier Factor n_neighbors=30, contamination=0.25	na	na	na	na	na
Local Outlier Factor n_neighbors=10	0,4	0,8	0,533333	0,6	0,48
Local Outlier Factor n_neighbors=30	0,4	0,8	0,533333	0,6	0,48
ocsvm gamma=0.01	0,4	0,8	0,533333	0,6	0,48
ocsvm gamma=0.03	na	na	na	na	na
ocsvm gamma=0.1 nu=0.01	0,1	1	0,181818	0,25	0,142857
ocsvm gamma=0.1 nu=0.13	0,5	0,833333	0,625	0,681818	0,576923
ocsvm gamma=0.1 nu=0.2	0,5	0,833333	0,625	0,681818	0,576923
ocsvm gamma=0.2	0,4	0,8	0,533333	0,6	0,48
ocsvm gamma=0.2 nu=0.01	0,1	1	0,181818	0,25	0,142857
ocsvm gamma=0.2 nu=0.2	na	na	na	na	na
ocsvm gamma=0.2 nu=0.3	na	na	na	na	na
ocsvm linear nu=0.1	0,4	0,8	0,533333	0,6	0,48
ocsvm linear nu=0.13	0,5	0,833333	0,625	0,681818	0,576923
ocsvm linear nu=0.15	na	na	na	na	na
ocsvm linear nu=0.25	na	na	na	na	na
One-class SVM default	0,4	0,8	0,533333	0,6	0,48

	Контролируемый				
	precision	recall	F0	F(prec)	F(rec)
Autoencoder	0,5	0,555556	0,526316	0,535714	0,517241
Elliptic Envelope contam=0.01	na	na	na	na	na
Elliptic Envelope contam=0.21	0,3	0,5	0,375	0,409091	0,346154
Elliptic Envelope contam=0.5	na	na	na	na	na
Elliptic Envelope default	0,2	0,666667	0,307692	0,375	0,26087
Isolation Frest contamination=0.01	na	na	na	na	na
Isolation Frest contamination=0.2	0,2	0,4	0,266667	0,3	0,24
Isolation Frest contamination=0.25	0,3	0,5	0,375	0,409091	0,346154
Isolation Frest contamination=0.3	0,3	0,5	0,375	0,409091	0,346154
Isolation Frest contamination=0.4	na	na	na	na	na
Isolation Frest contamination=0.5	0,5	0,454545	0,47619	0,46875	0,483871
Isolation Forest default	0,2	0,5	0,285714	0,333333	0,25
Local Outlier Factor default	0,1	0,333333	0,153846	0,1875	0,130435
Local Outlier Factor n_neighbors=30, contamination=0.25	0,2	0,4	0,266667	0,3	0,24
Local Outlier Factor n_neighbors=10	0,1	1	0,181818	0,25	0,142857
Local Outlier Factor n_neighbors=30	0,1	1	0,181818	0,25	0,142857
ocsvm gamma=0.01	0,2	0,4	0,266667	0,3	0,24
ocsvm gamma=0.03	na	na	na	na	na
ocsvm gamma=0.1 nu=0.01	0,2	0,333333	0,25	0,272727	0,230769
ocsvm gamma=0.1 nu=0.13	0,3	0,375	0,333333	0,346154	0,321429
ocsvm gamma=0.1 nu=0.2	0,3	0,5	0,375	0,409091	0,346154
ocsvm gamma=0.2	0,4	0,444444	0,421053	0,428571	0,413793
ocsvm gamma=0.2 nu=0.01	0,3	0,5	0,375	0,409091	0,346154
ocsvm gamma=0.2 nu=0.2	0,5	0,555556	0,526316	0,535714	0,517241
ocsvm gamma=0.2 nu=0.3	0,4	0,571429	0,470588	0,5	0,444444
ocsvm linear nu=0.1	0,1	1	0,181818	0,25	0,142857
ocsvm linear nu=0.13	na	na	na	na	na
ocsvm linear nu=0.15	0,3	0,75	0,428571	0,5	0,375
ocsvm linear nu=0.25	0,5	0,714286	0,588235	0,625	0,555556
One-class SVM default	0,3	0,375	0,333333	0,346154	0,321429

	Гибридный				
	precision	recall	F0	F(prec)	F(rec)
Autoencoder	0,5	0,555556	0,526316	0,535714	0,517241
Elliptic Envelope contam=0.01	na	na	na	na	na
Elliptic Envelope contam=0.21	na	na	na	na	na
Elliptic Envelope contam=0.5	0,9	0,5	0,642857	0,586957	0,710526
Elliptic Envelope default	na	na	na	na	na
Isolation Frest contamination=0.01	na	na	na	na	na
Isolation Frest contamination=0.2	0,1	1	0,181818	0,25	0,142857
Isolation Frest contamination=0.25	0,3	1	0,461538	0,5625	0,391304
Isolation Frest contamination=0.3	0,3	1	0,461538	0,5625	0,391304
Isolation Frest contamination=0.4	0,4	1	0,571429	0,666667	0,5
Isolation Frest contamination=0.5	0,4	0,8	0,533333	0,6	0,48
Isolation Forest default	0,1	1	0,181818	0,25	0,142857
Local Outlier Factor default	0,1	0,333333	0,153846	0,1875	0,130435
Local Outlier Factor n_neighbors=30, contamination=0.25	0,8	0,615385	0,695652	0,666667	0,727273
Local Outlier Factor n_neighbors=10	0,2	0,4	0,266667	0,3	0,24
Local Outlier Factor n_neighbors=30	0,2	0,5	0,285714	0,333333	0,25
ocsvm gamma=0.01	0,6	0,75	0,666667	0,692308	0,642857
ocsvm gamma=0.03	0,6	0,75	0,666667	0,692308	0,642857
ocsvm gamma=0.1 nu=0.01	na	na	na	na	na
ocsvm gamma=0.1 nu=0.13	na	na	na	na	na
ocsvm gamma=0.1 nu=0.2	na	na	na	na	na
ocsvm gamma=0.2	0,4	1	0,571429	0,666667	0,5
ocsvm gamma=0.2 nu=0.01	na	na	na	na	na
ocsvm gamma=0.2 nu=0.2	na	na	na	na	na
ocsvm gamma=0.2 nu=0.3	0,3	1	0,461538	0,5625	0,391304
ocsvm linear nu=0.1	0,1	1	0,181818	0,25	0,142857
ocsvm linear nu=0.13	0,2	1	0,333333	0,428571	0,272727
ocsvm linear nu=0.15	0,2	1	0,333333	0,428571	0,272727
ocsvm linear nu=0.25	0,2	0,5	0,285714	0,333333	0,25
One-class SVM default	0,5	0,714286	0,588235	0,625	0,555556

Список литературы

Асиновская Е. Ю. Проблема создания диалогового агента «самобранка» для обслуживания клиентов кафе и ресторанов : Выпускная квалификационная работа бакалавра филологии / Асиновская Е. Ю. — СПбГУ, 2016. — Рукопись.

Лукашевич Н. Тезаурусы в задачах информационного поиска. — Москва : Изд-во Московского университета, 2011.

Лукашевич Н., *Левчик А.* Создание лексикона оценочных слов русского языка РуСентиЛекс // Труды конференции OSTIS-2016. — С. 377—382.

Меньшиков И. Л., *Кудрявцев А. Г.* Обзор систем анализа тональности текста на русском языке // Молодой ученый. — 2012. — № 12. — С. 140—143. — URL: `https://moluch.ru/archive/47/5951/` (дата обр. 10.06.2018).

Хониева Е. А. Взаимодействие с интеллектуальной голосовой технологией: организация диалога и представления пользователей : Диссертация на соискание квалификационной степени магистра / Хониева Е. А. — СПбГУ, 2016. — Рукопись.

Barnabé-Lortie V. Active Learning for One-Class Classification : Thesis submitted to the Faculty of Graduate and Postdoctoral Studies in partial fulfillment of the requirements for the MCS degree in Computer Science / Barnabé-Lortie Vincent. — University of Ottawa, Canada, 2015.

Bell L., *Gustafson J.* Positive and Negative User Feedback in a Spoken Dialogue Corpus // Sixth International Conference on Spoken Language Processing. T. 1. — Beijing : International Speech Communication Association, 2000. — С. 589—592.

Brennan S. E., *Hulteen E. A.* Interaction and feedback in a spoken language system: A theoretical framework // Knowledge-Based Systems. — 1995. — Т. 8, № 2/3. — С. 143—151.

Breunig M. M., *Kriegel H. P.*, *Ng R. T.*, *Sander J.* LOF: identifying density-based local outliers // ACM sigmoid record. — 2000. — Май.

Chen Y., *Skiena S.* Building Sentiment Lexicons for All Major Languages // Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers) / под ред. K. Toutanova, H. Wu. — Baltimore : Association for Computational Linguistics, 2014. — С. 383—389.

Eckert W., *Levin E.*, *Pieraccini R.* Automatic evaluation of spoken dialogue systems // TWLT13: Formal semantics and pragmatics of dialogue. — 1998. — С. 99—110.

Estabrooks A., *Jo T.*, *Japkowicz N.* A Multiple Resampling Method for Learning from Imbalanced Data Sets // Computational Intelligence. — 2004. — № 20. — С. 18—36.

Gamon M. Sentiment classification on customer feedback data: noisy data, large feature vectors, and the role of linguistic analysis // Proceedings of the 20th international conference on Computational Linguistics. — Stroudsburg : Association for Computational Linguistics, 2004. — С. 841—847.

Gandhe S., *Traum D.* An evaluation understudy for dialogue coherence models // Proceedings of the 9th SIGdial Workshop on Discourse and Dialogue / под ред. D. Schlangen, B. A. Hockey. — Columbus : Association for Computational Linguistics, 2008. — С. 172—181.

Gass S. M. Input, interaction, and the second language learner. — 1997.

Griol D., *Hurtado L. F.*, *Segarra E.*, *Sanchis E.* A statistical approach to spoken dialog systems design and evaluation // Speech Communication. — 2008. — Т. 50, № 8. — С. 666—682.

Harris Z. S. Distributional structure // Word. — 1954. — Т. 10, № 2/3. — С. 146—162.

He H., *Garcia E. A.* Learning from imbalanced data // IEEE Transactions on knowledge and data engineering. — 2009. — Т. 21, № 9. — С. 1263—1284.

Higashinaka R., *Funakoshi K.*, *Araki M.*, *Tsukahara H.*, *Kobayashi Y.*, *Mizukami M.* Towards Taxonomy of Errors in Chat-oriented Dialogue Systems // Proceedings of the 16th Annual Meeting of the Special Interest Group on Discourse and Dialogue / под ред. A. Koller, G. Skantze, F. Jurcicek, M. Araki, C. Penstein Rose. — Prague : Association for Computational Linguistics, 2015. — С. 87—95.

Iwashita N. Negative feedback and positive evidence in task-based interaction: Differential effects on L2 development // Studies in Second Language Acquisition. — 2003. — Т. 25, № 1. — С. 1—36.

Japkowicz N. Supervised versus unsupervised binary-learning by feedforward neural networks // Machine Learning. — 2001. — Т. 42, № 1/2. — С. 97—122.

Kay P., *Fillmore C. J.* Grammatical constructions and linguistic generalizations: the What's X doing Y? construction // Language. — 1999. — Т. 75, № 1. — С. 1—33.

Keizer S., *Bunt H.* Evaluating combinations of dialogue acts for generation // Proceedings of the 8th SIGdial Workshop on Discourse and Dialogue / под ред. S. Keizer, H. Bunt, T. Paek. — Antwerp : Association for Computational Linguistics, 2007. — С. 158—165.

Lau J. H., *Baldwin T.* An empirical evaluation of doc2vec with practical insights into document embedding generation // CoRR. — 2016. — arXiv preprint: arXiv:1607.05368. — (Дата обр. 10.06.2018).

Laurikkala J. Improving Identification of Difficult Small Classes by Balancing Class Distribution // Proceedings of the Conference AI in Medicine in Europe: Artificial Intelligence Medicine. — 2001. — С. 63—66.

Le Q., *Mikolov T.* Distributed representations of sentences and documents // Proceedings of the 31st International Conference on International Conference on Machine Learning. Т. 32. Ч. 2. — Beijing, 2014. — С. 1188—1196.

Litman D. J., *Pan S.* Empirically Evaluating an Adaptable Spoken Dialogue System // UM '99: Proceedings of the seventh International Conference on User Modeling / под ред. J. Kay. — New-York : Springer, 1999. — С. 55—64.

Litman D. J., *Pan S.*, *Walker M. A.* Evaluating Response Strategies in a Web-Based Spoken Dialogue Agent // Proceedings of the 17th international conference on Computational linguistics. Т. 2 / под ред. C. Boitet, P. Whitelock. — Stroudsburg : Association for Computational Linguistics, 1998. — С. 780—786.

Liu C.-W., *Lowe R.*, *Serban I. V.*, *Noseworthy M.*, *Charlin L.*, *Pineau J.* How NOT To Evaluate Your Dialogue System: An Empirical Study of Unsupervised Evaluation Metrics for Dialogue Response Generation // CoRR. — 2016. — arXiv preprint: arXiv:1603.08023. — (Дата обр. 10.06.2018).

Liu F. T., *Ting K. M.*, *Zhou Z.-H.* "Isolation forest." Data Mining // ICDM'08. Eighth IEEE International Conference. — 2008.

Lughofer E. Hybrid active learning for reducing the annotation effort of operators in classification systems. Pattern Recognition. — 2012.

Maslova N., *Potapov V.* Neural Network Doc2vec in Automated Sentiment Analysis for Short Informal Texts // 19th International Conference on Speech and Computer / под ред. A. Karpov, R. Potapova, I. Mporas. — Cham : Springer, 2017. — С. 546—554.

Mena L., *Gonzalez J. A.* Symbolic one-class learning from imbalanced datasets: application in medical diagnosis // International Journal on Artificial Intelligence Tools. — 2009. — Т. 18, № 02. — С. 273—309.

Mescheryakova E. I., *Nesterenko L. V.* Domain-Independent Classification of Automatic Speech Recognition Texts // Computational Linguistics and Intellectual Technologies: papers from the Annual conference “Dialogue”. Т. 16 / под ред. V. P. Selegey. — Moscow : RSUH, 2017. — С. 146—154.

Mikolov T., *Chen K.*, *Corrado G.*, *Dean J.* Efficient estimation of word representations in vector space // CoRR. — 2013. — arXiv preprint: arXiv:1301.3781. — (Дата обр. 10.06.2018).

Möller S. Parameters for Quantifying the Interaction with Spoken Dialogue Telephone Services // 6th SIGdial Workshop on Discourse and Dialogue / под ред. L. Dybkjær, W. Minker. — Lisbon : International Speech Communication Association, 2005.

Paek T. Empirical Methods for Evaluating Dialog Systems // Proceedings of the ACL 2001 Workshop on Evaluation Methodologies for Language and Dialogue Systems. Т. 9 / под ред. P. Paroubek. — Stroudsburg : Association for Computational Linguistics, 2001. — С. 1—9.

QasemiZadeh B., *Kallmeyer L.*, *Passban P.* Sketching Word Vectors Through Hashing // CoRR. — 2017. — arXiv preprint: arXiv:1705.04253. — (Дата обр. 10.06.2018).

Rousseeuw P., *Van Driessen K.* A fast algorithm for the minimum covariance determinant estimator // Technometrics. — 1999. — Т. 41, № 3.

Schölkopf B. Estimating the support of a high-dimensional distribution // Neural computation. — 2001. — Т. 13, № 7.

Sun Y. Cost-sensitive boosting for classification of imbalanced data : A thesis presented to the University of Waterloo in fulfilment of the thesis requirement for the degree of Doctor of Philosophy / Sun Yanmin. — University of Waterloo, Canada, 2007.

Walker M. A., *Litman D. J.*, *Kamm C. A.*, *Abella A.* PARADISE: A Framework for Evaluating Spoken Dialogue Agents // Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and Eighth Conference of the European Chapter of the Association for Computational Linguistics / под ред. P. R. Cohen, W. Wahlster. — Stroudsburg : Association for Computational Linguistics, 1997. — С. 271—280.

Weiss G. M., *Provost F.* The Effect of Class Distribution on Classifier Learning: An Empirical Study // Technical Report MLTR- 43. — Dept. of Computer Science, Rutgers University, 2001.

Yang Z., *Levow G. A.*, *Meng H.* Predicting user satisfaction in spoken dialog system evaluation with collaborative filtering // IEEE Journal of Selected Topics in Signal Processing. — 2012. — Т. 6, № 8. — С. 971—981.