

<https://doi.org/10.21869/2223-1560-2019-23-3-86-99>

## Алгоритмы автоматизированного обучения диалоговых систем

Д.В. Спирин ✉<sup>1</sup>, О.С. Брежнев<sup>1</sup>

<sup>1</sup> Пензенский государственный университет, ул. Красная, 40, г. Пенза, 440026, Российская Федерация

✉ e-mail: spirin.dmitrij@list.ru

### Резюме

**Цель исследования.** Представленное в данной статье исследование проведено в рамках проекта Salebot.pro (на ресурсе <https://salebot.pro>) и было нацелено на разработку простой и эффективной реализации диалоговой системы.

**Методы.** План исследования предусматривал анализ различных методов обработки естественных языков и машинного обучения. Реализации методов были взяты из популярных библиотек с открытым исходным кодом. Построена модель диалоговой системы в двух вариантах: на основе фреймворка Spacy и метрического алгоритма оценки, на основе расстояния Левенштейна. Сравнивались простота реализации и затраты на обучение системы и персонала.

**Результаты.** Описанные в статье алгоритмы сопоставляют наиболее похожие слова из двух текстов и подсчитывают средний процент совпадений. Такой подход обеспечивает возможность приемлемой работы на языках со свободным порядком слов, к которым относится и русский язык. Выполненное исследование позволило разработать алгоритм автоматизированного обучения диалоговых систем в режиме реального времени без потери контекста. На той же основе разработан алгоритм обучения диалоговой системы по истории диалога. Предлагается использовать данные алгоритмы совместно. При создании диалоговой системы первоначально необходимо ее обучить на истории диалогов, а затем перманентно обучать в режиме реального времени.

**Заключение.** Достоинством разработанного алгоритма является легкость в реализации и дешевизна построения инфраструктуры, необходимой для обучения модели, и ее обслуживания, а также простота в эксплуатации. Применяется подход, который отличается от обучения с учителем, что позволяет ускорить процесс обучения и ввода в систему новых данных. Особенностью разработанных алгоритмов является игнорирование семантики текста, что делает обучение автоматизированным, а не автоматическим.

---

**Ключевые слова:** диалоговая система; конечный автомат; фрейм; автоматизированное обучение; алгоритм.

**Конфликт интересов:** Авторы декларируют отсутствие явных и потенциальных конфликтов интересов, связанных с публикацией настоящей статьи.

**Для цитирования:** Спирин Д.В., Брежнев О.С. Алгоритмы автоматизированного обучения диалоговых систем // Известия Юго-Западного государственного университета. 2019; 23(3): 86-99. <https://doi.org/10.21869/2223-1560-2019-23-3-86-99>.

Статья поступила в редакцию 17.05.2019

Статья подписана в печать 05.06.2019

<https://doi.org/10.21869/2223-1560-2019-23-3-86-99>

## Automated Training Algorithms of Dialog Systems

Dmitriy V. Spirin<sup>1</sup>, Oleg S. Brezhnev<sup>1</sup>

<sup>1</sup> Penza State University, 40, Krasnaya str., Penza, 440026, Russian Federation

✉ e-mail: spirin.dmitrij@list.ru

### Abstract

**Purpose of research.** The research described in this article is conducted within the Salebot.pro project (on the <https://salebot.pro> resource) and aimed at development of simple and effective realization of a dialog system.

**Methods.** The research plan provided the analysis of various methods of natural processing languages and machine learning languages. Implementation of these methods was taken from popular libraries with an open source code. The model of a dialog system was made in two options: on the basis of Spacy freymvork and metric assessment algorithm, on the basis of Levenstein's distance. Simplicity of implementation and costs on training of a system and personnel were compared.

**Results.** The algorithms described in article compare the most similar words from two texts and count average percent of coincidence. Such approach provides a possibility of acceptable work in languages with free word order. Russian is one such languages. The executed research allowed developing an automated training algorithm of dialog systems in real time without context loss. On the same basis training algorithm of a dialog system in dialog history is developed. It is offered to use these algorithms together. It is originally necessary to train it at history of dialogues during creation of a dialogue system. And then it is necessary to train it permanently in real time.

**Conclusion.** The advantage of the developed algorithm is ease in implementation and low cost of infrastructure which is necessary for model training and its service and also operation simplicity. Approach which differs from training with the teacher allows accelerating training process and input of new data into the system. Specific feature of the developed algorithms is ignoring of text semantics that makes training automated but not automatic.

**Keywords:** dialog system; finite-state machine; frame; automated training; algorithm.

**Conflict of interest:** The Authors declare the absence of obvious and potential conflicts of interest related to the publication of this article.

**For citation:** Spirin D. V., Brezhnev O.S. Automated Training Algorithms of Dialog Systems. *Izvestiya Yugo-Zapadnogo gosudarstvennogo universiteta = Proceedings of the Southwest State University*. 2019, 23(3): 86-99 (In Russ.). <https://doi.org/10.21869/2223-1560-2019-23-3-86-99>.

Received 12.03.2019

Accepted 04.04.2019

\*\*\*

### Введение

Разработкой систем виртуального общения многие специалисты занима-

ются уже многие годы. На сегодняшний день исследования в данной области

весьма актуальны из-за обширных возможностей коммерциализации различных видов человеческой деятельности. Система может заменить как операторов технической поддержки, так и менеджеров по продажам. Диалоговые системы могут разрабатываться как с применением методов искусственного интеллекта, так и без них. Проблемой при разработке диалоговых систем является сложность их обучения; оно требует значительного количества вычислительных ресурсов, а также большого объема обучающих данных [1]. При обучении также не учитывается контекст и смысл реплик, что временами приводит к неожиданным результатам. Поэтому в рамках представленного исследования была поставлена цель – разработать простую архитектуру для быстрых и эффективных диалоговых систем.

Поведение виртуального собеседника можно описать автоматной моделью с соответствующими состояниями. Контекстом подобного диалога является как история общения, так и все состояния, через которые прошел собеседник в результате общения с виртуальной системой. Каждое состояние автомата возможно представить фреймом, полно описывающим текущее состояние и возможные переходы. Это системы с закрытой областью, ориентированные на решение задачи, так как фреймы описывают полно текущее состояние, и невозможно описать все возможные ситуации диалога. Рассматри-

ваемые алгоритмы обучения дотраивают автомат, по которому происходило общение в диалоге.

### Материалы и методы решения задачи

Основой при разработке описываемых далее алгоритмов является определение подходящих состояний автомата для последующего сопоставления текстов на естественных языках, что является нетривиальной задачей. Структура естественных языков допускает множество неоднозначностей при попытке определить смысл текста при помощи алгоритмических методов, поэтому анализ предложений является трудоемким процессом.

Для работы с текстом используют специальные библиотеки, такие как spaCy [2] или OpenNLP [3]. Программные компоненты содержат в своем составе широкий спектр инструментов для работы с текстом в процессе обработки естественных языков: уже готовые токенизаторы, синтаксические анализаторы и многое другое. Проблема состоит в том, что для качественной работы программных компонент необходим большой объем данных. Такие программные инструменты требуют ручного обучения и тяжело контролируются, что влечет дополнительные расходы на содержание специалистов, способных построить и сопровождать инфраструктуру для автоматизации отделов технической поддержки. Функционал часто избыточен, если требуется только сравнение текста. В этом случае возможно

использовать алгоритм, основанный на подсчете редакционного расстояния (расстояния Левенштейна).

Описанные в статье алгоритмы сопоставляют наиболее похожие слова из двух текстов и подсчитывают средний процент совпадений. Такой подход обеспечивает возможность приемлемой работы на языках со свободным порядком слов, к которым относится и русский язык.

Задачу сопоставления текстов можно разбить на несколько подзадач. Первоначально требуется избавиться от лишней неоднозначности в тексте, не очень сильно влияющей на смысл текста – то есть нормализовать текст. Следующим шагом нужно дать оценку схожести произвольных строк – это и есть алгоритм подсчета редакционного расстояния с небольшими изменениями. На заключительном этапе проводится сопоставление наиболее схожих слов на основе оценок схожести – это задача о назначениях, и решать ее в описанных ниже алгоритмах предложено при помощи Венгерского алгоритма, сложность которого полиномиальна.

Венгерский алгоритм легко расширяется на случай решения проблемы оценки расстояний для текстов с различным количеством слов. Для слов без пары необходимо принять за редакционное состояние количество символов, входящих в это слово. Таким образом алгоритм упрощается без потери качества обработки. В процессе исследования было проведено сравнение производительности трех реализаций: комбинаторного для предложений, комбина-

торного для слов и Венгерского алгоритмов.

Комбинаторная реализация сопоставления по словам показывает неудовлетворительное качество для длинных предложений, однако является устойчивой к опечаткам и показывает хороший результат. Асимптотика данного алгоритма представляется факториальной функцией, что делает его неприемлемым для практического использования [4]. Сопоставление по предложениям не отвечает требованиям качества, поскольку очень высок процент ложных срабатываний, сложность остается факториальной, но алгоритм может работать для текстов больших объемов. Венгерский алгоритм позволил обойти ограничение по времени и обрабатывать тексты большого объема без потери качества.

#### Алгоритм обучения по истории диалогов

Диалоговая система, которая заменяет оператора, может быть обучена на истории диалогов. Алгоритм обучения выявляет деревья в диалогах и создает эквивалентный автомат. Во время работы алгоритм пытается минимизировать количество деревьев и состояний: происходит поиск по уже имеющимся состояниям. Несколько подряд идущих сообщений от оператора или собеседника соединяются в одно сообщение, если интервал времени между ними небольшой (промежуток времени задается отдельно). Необходимость такого объединения продиктована тем, что люди при переписке в реальном времени часто могут разбивать свои мысли на не-

сколько сообщений, хотя эти сообщения относятся к одному контексту [5].

На рис. 1 представлена схема алгоритма проверки существования в автомате подходящего состояния для реплики.

Алгоритм работает со всей историей и можно принять, что первым сообщением всегда является начальное. Сначала проводится поиск корня, удовлетворяющего условию.

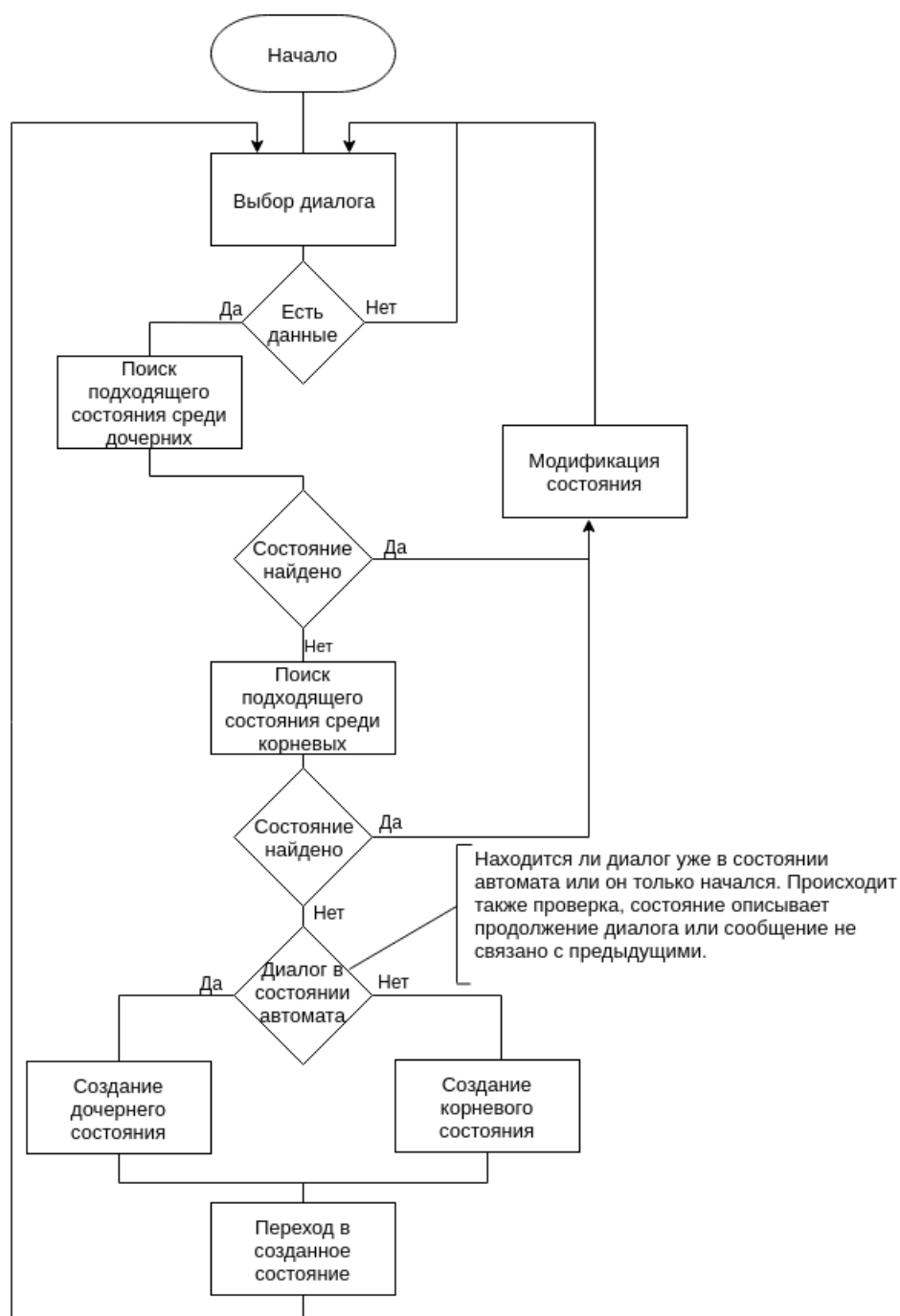


Рис. 1. Схема алгоритма обучения по истории диалогов

Fig. 1. Scheme of learning algorithm on the history of dialogues

Если корень найден, то следующее сообщение будет проверяться только среди узлов, дочерних тому, в которое они вошли, иначе создается новое дерево и проверок больше не происходит.

В тех случаях, когда такое состояние не найдено, оно будет создано диалоговой системой. Если подходящее состояние уже существует, но редакционное расстояние не равно нулю, то происходит его модификация с учетом информации, полученной из текущей реплики. Если же имеется полностью эквивалентное состояние, то никаких операций производить не требуется.

Алгоритм выполняется раз за разом для каждого нового диалога и для каждой реплики в нем. В результате работы алгоритма должен получиться лес из деревьев, каждое дерево которого разветвляется на возможные варианты развития диалогов.

#### Алгоритм обучения в режиме реального времени

Современные диалоговые системы в текущем состоянии их развития не способны заменить человека с приемлемым качеством реплик, поэтому алгоритм предполагает совместную работу диалоговой системы и оператора. Оператор будет контролировать процесс обучения и результаты.

Диалоговая система должна взять на себя взаимодействие с пользователем по типовым вопросам, которые предусмотрены сценарием. По мере обучения системы сценарий будет расширяться, и роль оператора будет по-

степенно уменьшаться. Оператор также выполняет роль модератора, в тех случаях, когда не используется какая-либо автоматизированная система модерации входящих сообщений.

В тот момент, когда бот не знает ответа на реплику пользователя, в диалог включается оператор. Диалоговая система анализирует реплику пользователя и ответ на нее оператора. На основе этих данных создается дополнительное состояние в автомате. Для того, чтобы не потерять контекст диалога, состояние должно быть связано с предыдущим и не является начальным. Если реплика является первой в диалоге, то будет создано начальное состояние автомата. После создания нового состояния конечный автомат переходит в это состояние. Затем цикл обучения повторяется до тех пор, пока диалог не вернется в уже известное состояние или не будет инициирован с самого начала.

Вопрос, который не требует понимания контекста диалога, можно отметить как начальный узел диалога. Тогда вопросы, которые никак не связаны с текущим контекстом, будут обрабатываться в любой точке диалога [6]. Существует возможность улучшить реализацию с помощью конечного автомата с магазинной памятью и ограниченным размером стека для переключения между контекстами диалога. Возможность запоминания контекста позволит возвращаться к прерванным диалогам в случае переключения контекста. Соответствующая схема алгоритма представлена на рисунке 2.

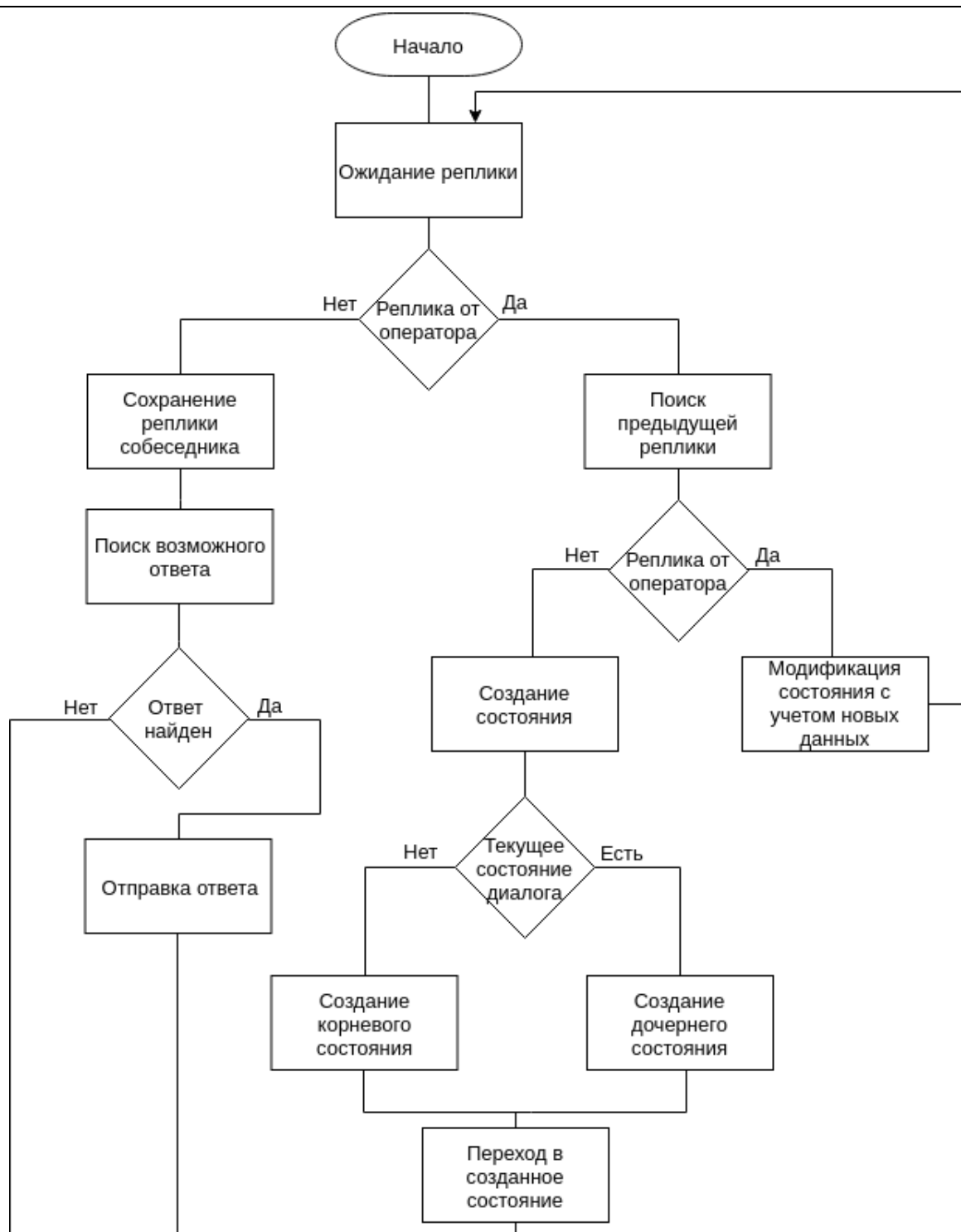


Рис. 2. Схема алгоритма обучения в режиме реального времени

Fig. 2. Diagram of the learning algorithm in real-time

Из схемы видно, что обучение работает только случае, когда в диалог включается оператор. В остальное время обучения не происходит. Это позво-

ляет сократить время на обработку входящих сообщений, сохранив при этом качество обучения.

## Результат обучения

Алгоритмы реализованы на языке Python с использованием модуля для подсчета редакционного расстояния и сравнения строк. Результаты были про-

тестированы на реальных данных истории диалогов отдела технической поддержки.

Ниже представлена программная реализация алгоритма обучения по истории диалога:

```
def teach_by_history(self):
    for dialog in self.dialogs:
        prev_replica = None
        prev_state = None
        answer = ''
        condition = ''
        l = len(dialog)
        for index, replica in enumerate(dialog):
            self.processed += 1
            if len(replica['message']) == 0:
                continue
            if replica['inbox']:
                if condition == '':
                    condition = replica['message']
                elif prev_replica != None and self.is_different_dialogs(prev_replica, replica):
                    condition = replica['message']
                prev_state = None
                m = self.find_state('#{none}', prev_replica['message'], x, y, prev_state, True)
            else:
                condition += ' ' + replica['message']
        else:
            if index < (l - 1):
                next_replica = dialog[index + 1]
            else:
                next_replica = None
            if prev_state == None or self.is_different_dialogs(prev_replica, replica):
                if next_replica != None and next_replica['inbox']:
                    answer = replica['message']
                    m = self.find_state(answer, condition, x, y, prev_state, True)
                    prev_state = m
                    condition = ''
                else:
                    if next_replica != None and next_replica['inbox']:
                        # создание соединения
                        answer = replica['message']
                        m = self.find_state(answer, condition, x, y, prev_state, False)
                        prev_state = m
                        condition = ''
                    else:
                        # соединить ответы
                        answer += ' ' + replica['message']
        prev_replica = replica
```



При работе алгоритма осуществляется перебор всех состояний, в которые может быть осуществлен переход, поэтому с увеличением размера обучающей выборки время обучения на каждом последующем диалоге возрастает. Сложность алгоритма стремится к  $O(n)$ , где  $n$  количество начальных состояний автомата. Для сложных диалоговых систем необходимо усовершенствовать реализацию, применив хеширование к сортировке состояний диалога для ускорения поиска подходящего перехода.

Для небольших фреймов с количеством состояний, не превышающим 250, время перебора незначительно относительно времени работы алгоритма сопоставления текстов в узлах состояний. Наиболее затратной операцией является сопоставление слов в текстах.

Такая сложность позволяет строить диалоговые системы на компьютерах среднего ценового диапазона и не требует какого-либо специального оборудования для обучения. Такая нетребовательность к ресурсам расширяет сферу применения диалоговых систем и дает возможность использовать их для нужд малого бизнеса, то есть является экономически выгодным решением: так поиск состояния в диалоговой системе из 150 узлов с самым длинным текстом из 1000 символов займет  $15 \cdot 108$  операций, что займет одну-две секунды на процессоре Intel Pentium 4 (6 GFlops) [7].

На рисунке 3 представлена форма отображения результата обучения системы по истории диалога с ручным форматированием расположения блоков для наибольшей наглядности.

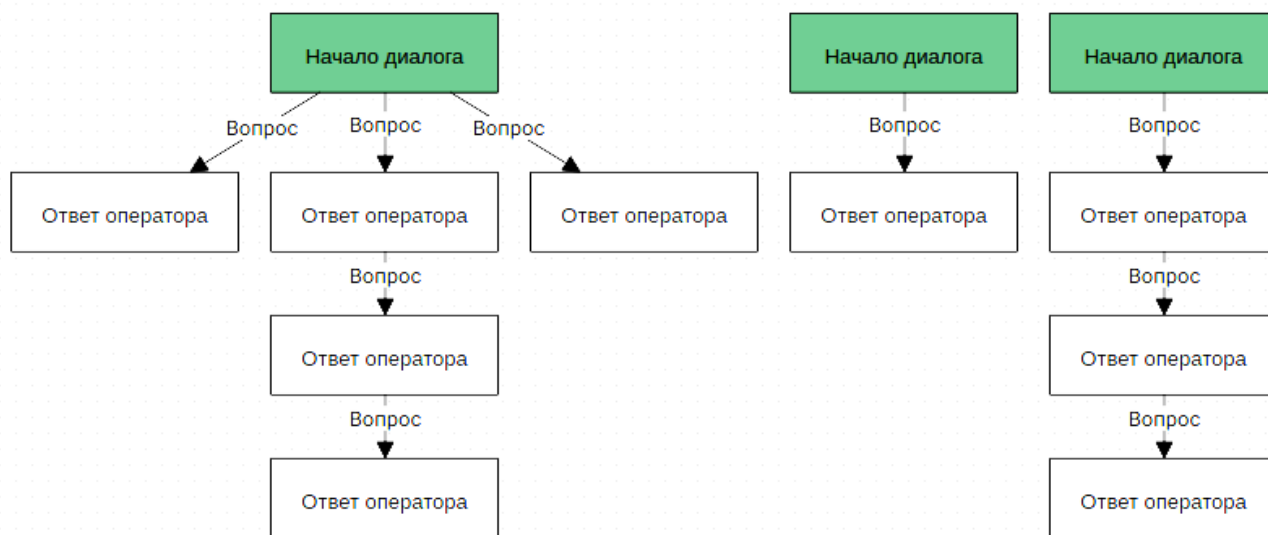


Рис. 3. Форма отображения результата обучения системы

Fig. 3. The form of displaying the result of training system

По рисунку 3 видно, что велись несвязанные диалоги, так как объединение деревьев минимально. Темным цветом показаны начальные состояния ав-

томата, блоки белого цвета отображают внутренние состояния автомата, стрелки – переходы между состояниями. Каждый диалог представляет собой ли-

нейную последовательность блоков один под другим. Одинаковые диалоги объединились между собой с разветвлениями в местах, где существовали различия.

Представление диалога в виде блок-схемы закладывает возможность дальнейшего расширения диалоговой системы за счет модификации результатов человеком, ответственным за обучение

системы. Помимо этого, такое представление позволяет тонко настроить поведение системы, что улучшает опыт взаимодействия пользователя с системой.

На рисунке 4 представлена часть реального результата обучения системы по истории диалога с ручным форматированием расположения блоков для наибольшей наглядности.

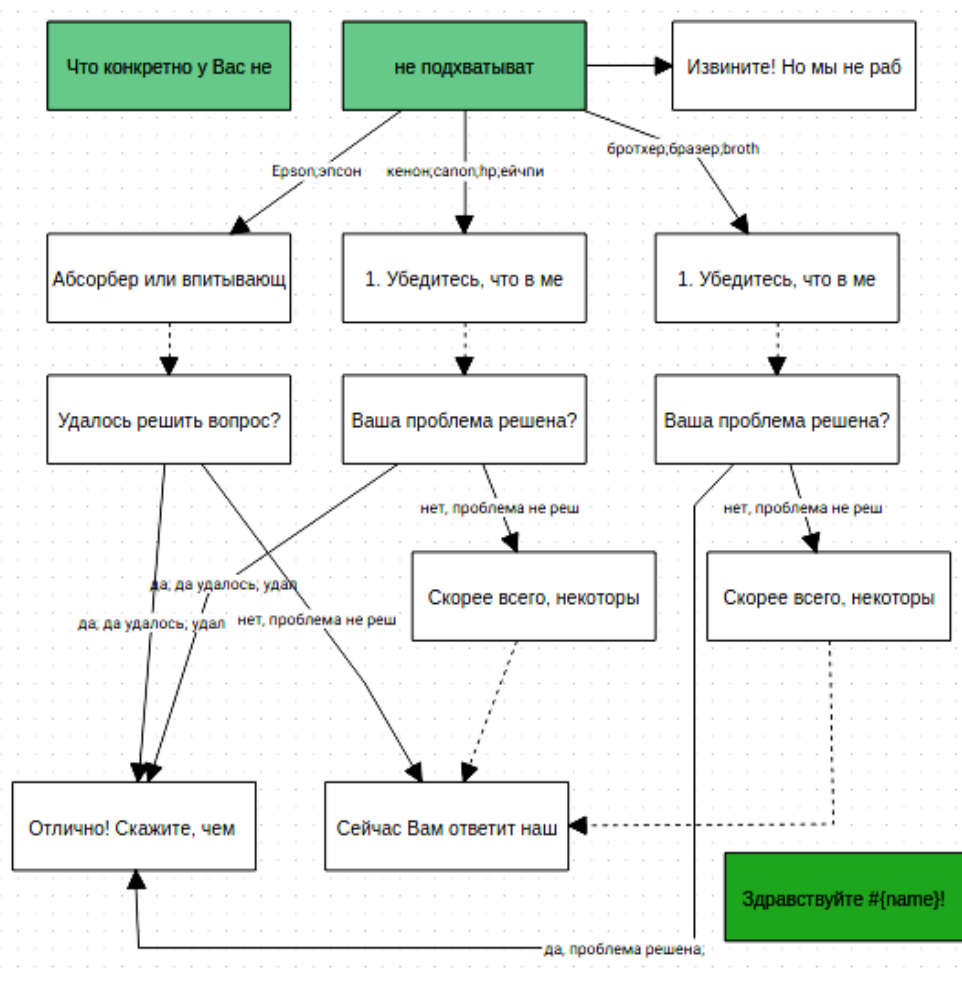


Рис. 4. Реальный результат обучения системы (снимок с экрана)

Fig. 4. Real system learning result (screenshot)

Следует отметить, что тестирование проходило на реальных диалогах и схема включает оригинальную орфо-

графию. Схема построена автоматически реализацией алгоритма обучения на ресурсе <https://salebot.pro>. Рисунок 4 яв-

ляется снимком экрана рабочей поверхности системы разработки диалоговых систем Salebot.pro, в которой был реализован предложенный алгоритм автоматизированного обучения. Текст реплик в блоках на рисунке 4 отображается не полностью, поэтому для просмотра необходимо использовать курсор.

Тестирование результатов обучения на реальных диалогах подтвердило жизнеспособность подхода и его преимущества по сравнению с обучением на предварительно размеченных данных. В случае с обучением на реальных данных сокращается время работы над моделью – изменения можно применять сразу после проверки модели на адекватность человеком, причем оператору не требуется специальная квалификация [8]. С этой стороны реализуемый подход показал свое преимущество перед методами обучения с учителем (подход, используемый Spacy и OpenNLP).

### Результаты и их обсуждения

Алгоритмы обучения значительно упрощают создание диалоговых систем. Описанные алгоритмы являются автоматизированными, а не автоматическими, так как после обучения требуется ручная проверка результатов, поскольку они не способны определить, какое из создаваемых состояний может быть начальным, так как они не анализируют контекст. За начальные состояния принимаются только те из них, с которых начинался диалог. Если количество диалогов значительно, а их глубина невелика, то алгоритм показывает лучшие

результаты обучения. После обучения необходимо в ручном режиме проверить результаты и выбрать те состояния диалога, которые не связаны с предыдущими репликами. Такие состояния могут быть назначены начальными состояниями диалога.

В процессе выполнения алгоритм запоминает все ошибки, опечатки и неточности, которые были допущены оператором. Все они попадут в схему диалога и их необходимо будет исправлять. К примеру, если в диалоге фигурировало имя собеседника, то обращение в других диалогах будет также по имени, поэтому после обучения необходимо выделить изменяемые сущности и значения, получаемые извне в ручном режиме. Подобные ситуации также можно исправить с помощью подходов NER.

Обучение идет только по истории диалога, фактор творчества отсутствует, следовательно, пока алгоритм не встретит неизвестную ситуацию, он не научится ее разрешать. При работе диалоговой системы совместно с оператором данная особенность не существенна и со временем участие оператора требуется все реже. Один оператор может отвечать за множество диалоговых систем, что уменьшает штат сотрудников, требуемый для обслуживания диалоговых систем.

Положительной стороной данного алгоритма является то, что он не является “черным ящиком” – на него можно повлиять, представив автомат в виде схемы, состоящей из блоков, соединенных связями. Представленная особен-

ность является решающей в выборе данного алгоритма к использованию. Вероятность худшего случая, когда алгоритм обучения может навредить обучаемой системе, минимальна, так как в нем только повторяются действия оператора, и весь результат обучения виден и может быть либо исправлен, либо отменен.

Во время обучения все реплики собеседника сохраняются в состояниях. Это порождает ряд проблем – необходимо исключать из обучения реплики, содержащие конфиденциальную информацию. Для этого можно использовать методы распознавания именованных сущностей NER (Named Entity Recognition) [9, 10, 11], либо ручную модерацию. Представляет большой интерес интеграция предложенного в настоящей работе подхода с методами NER, реализованными в нейроподобных структурах [12, 13], в том числе на основе использования метода глубокого обучения искусственных нейронных сетей [14], что позволит повысить эффек-

тивность взаимодействия с диалоговой системой.

## Выводы

Предложенный в настоящей работе подход, основанный на обучении без учителя, является гибким и позволяет достигнуть наилучших результатов в совокупности с другими подходами. У предложенного алгоритма имеется большой потенциал для расширения, что является его важным преимуществом. На основании проведенного анализа предлагается использовать предложенные алгоритмы совместно. На первом этапе создания диалоговой системы необходимо проводить ее обучение на истории диалогов или заранее написанном сценарии, а затем непрерывно обучать в режиме реального времени. Предполагается, что дальнейшее внедрение последних достижений в области искусственных нейронных сетей при распознавании именованных сущностей позволит получить простые и эффективные реализации диалоговых систем.

## Список литературы

1. Провотар А.И., Ключко К. А. Особенности и проблемы виртуального общения с помощью чат-ботов // Информационные технологии и компьютерная техника. Научные работы ВНТУ. 2013. № 3. С. 1-6.
2. Training spaCy's Statistical Models. URL: <https://spacy.io/usage/training> (дата обращения: 07.05.2019).
3. Apache OpenNLP Developer Documentation. URL: <https://opennlp.apache.org/docs/1.9.0/manual/opennlp.html> (дата обращения: 07.05.2019).
4. Задача о редакционном расстоянии, алгоритм Вагнера-Фишера. URL: [https://neerc.ifmo.ru/wiki/index.php?title=Задача\\_о\\_редакционном\\_расстоянии,\\_алгоритм\\_Вагнера-Фишера](https://neerc.ifmo.ru/wiki/index.php?title=Задача_о_редакционном_расстоянии,_алгоритм_Вагнера-Фишера) (дата обращения: 07.05.2019).

5. Ramsay A. Discourse. In Mitkov, R. (Ed.). *The Oxford Handbook of Computational Linguistics*. Oxford University Press, USA, 2003. 717 p.
6. Traum D., Larsson S. The information state approach to dialogue management // In J. van Kuppevelt & R. Smith (Eds.), *Current and new directions in discourse and dialogue*. Springer, 2003. P. 325–354.
7. Computing Power Throughout History. URL: [https://www.alternatewars.com/BBOW/Computing/Computing\\_Power.htm](https://www.alternatewars.com/BBOW/Computing/Computing_Power.htm) (дата обращения: 07.05.2019).
8. Автоматизированное обучение. URL: <https://salebot.pro/articles/9> (дата обращения: 07.05.2019).
9. Спирин Д.В., Брежнев О.С., Баринев А.Д. Алгоритм автоматизированного обучения // Сборник статей II Международной научно-практической конференции. Пенза: МЦНС «Наука и Просвещение», 2018. С. 49-53.
10. A multi-task approach for named entity recognition in social media data / G. Aguilar, S. Maharjan, A. Pastor Lopez-Monroy, T. Solorio // In *Proceedings of the 3rd Workshop on Noisy User-generated Text*, 2017. P. 148–153.
11. Daniken P., Cieliebak M. Transfer learning and sentence level features for named entity recognition on tweets // In *Proceedings of the 3rd Workshop on Noisy User-generated Text*, 2017. P. 166–171.
12. Neural Architectures for Named Entity Recognition / G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, C. Dyer // In *Proceedings of NAACL-HLT 2016*, San Diego, California, June 12-17, 2016. P. 260–270.
13. Strakova J. *Neural Network Based Named Entity Recognition*. – Institute of Formal and Applied Linguistics, Prague. 2017. 120 p.
14. Akkaya E.K. *Deep neural networks for named entity recognition on social media*. Computer Engineering Dept., Hacettepe University. Beytepe-Ankara, Turkey, 2018. 126 p.

## Reference

1. Provotar A. I., Klochko K. A. Osobennosti i problemy virtual'nogo obshcheniya s pomoshch'yu chat-botov [Features and problems of virtual communication using chat bots]. *Informatsionnye tekhnologii i komp'yuternaya tekhnika. Nauchnye raboty VNTU. = Information technologies and computer equipment Scientific works VNTU*, 2013, no. 3, pp. 1-6 (In Russ.).
2. [Training spaCy's Statistical Models]. Available at: <https://spacy.io/usage/training> (accessed 07.05.2019).
3. Apache OpenNLP Developer Documentation. Available at: <https://opennlp.apache.org/docs/1.9.0/manual/opennlp.html> (accessed 07.05.2019).
4. Zadacha o redaktsionnom rasstoyanii, algoritm Vagnera-Fishera [The task of the editorial distance, the algorithm of Wagner-Fisher]. Available at: The access method is free: [https://neerc.ifmo.ru/wiki/index.php?title=Task\\_about\\_education\\_distance,\\_algorithm\\_Wagner-Fisher](https://neerc.ifmo.ru/wiki/index.php?title=Task_about_education_distance,_algorithm_Wagner-Fisher) (accessed 07.05.2019) (In Russ.).

5. Ramsay A. Discourse. In Mitkov, R. (Ed.). The Oxford Handbook of Computational Linguistics. Oxford University Press, USA, 2003, 717 p.
6. Traum D., Larsson S. The information state approach to dialogue management. In J. van Kuppevelt & R. Smith (Eds.), Current and new directions in discourse and dialogue Springer, 2003, p. 325–354.
7. Computing Power Throughout History Available at: [https:// www.alternatewars.com/BBOW/ Computing / Computing\\_Power.htm](https://www.alternatewars.com/BBOW/Computing/Computing_Power.htm) (accessed 07.05.2019).
8. Avtomatizirovannoe obuchenie. Available at: The access method is free: <https://salebot.pro/articles/9> (accessed 07.05.2019) (In Russ.).
9. Spirin D.V., Brezhnev O. S., Barinov A. D. [Algorithm of automated learning]. *Sbornik statei II Mezhdunarodnoi nauchno-prakticheskoi konferentsii* [Collection of articles of the II International Scientific and Practical Conference]. Penza, 2018, pp. 49-53 (In Russ.).
10. Aguilar G., Maharjan S., Pastor Lopez-Monroy A., Solorio T..A multi-task approach for named entity recognition in social media data. In Proceedings of the 3rd Workshop on Noisy User-generated Text, 2017, pp. 148–153.
11. Daniken P., Cieliebak M. Transfer learning and sentence level features for named entity recognition on tweets. In Proceedings of the 3rd Workshop on Noisy User-generated Text, 2017, pp. 166–171.
12. Lample G., Ballesteros M., S Subramanian., Kawakami K., Dyer C. Neural Architectures for Named Entity Recognition. In Proceedings of NAACL-HLT 2016, San Diego, California, June 12-17, 2016, pp. 260–270.
13. Strakova J. Neural Network Based Named Entity Recognition. Institute of Formal and Applied Linguistics, Prague, 2017, 120 p.
14. Akkaya E.K. Deep neural networks for named entity recognition on social media. Computer Engineering Dept., Hacettepe University, Beytepe-Ankara, Turkey, 2018. 126 p.

---

### Информация об авторах / Information about the Authors

**Дмитрий Владимирович Спирин**, аспирант кафедры вычислительной техники, Пензенский государственный университет, г. Пенза, Российская Федерация,  
e-mail: [spirin.dmitrij@list.ru](mailto:spirin.dmitrij@list.ru)

**Dmitriy V. Spirin**, Post-Graduate Student, Department of Computing Engineering, Penza State University, Penza, Russian Federation  
e-mail: [spirin.dmitrij@list.ru](mailto:spirin.dmitrij@list.ru)

**Олег Сергеевич Брежнев**, инженер-программист кафедры математического обеспечения и применения ЭВМ, Пензенский государственный университет, г. Пенза, Российская Федерация,  
e-mail: [oleg-423@yandex.ru](mailto:oleg-423@yandex.ru)

**Oleg S. Brezhnev**, Software Engineer, Department of Software and Computer Applications, Penza, Russian Federation  
e-mail: [oleg-423@yandex.ru](mailto:oleg-423@yandex.ru)