

Интерфейс на основе нейросетевых языковых моделей для взаимодействия человека и робота на русском языке

А. Г. Сбоев, email: Sboev_AG@nrcki.ru ^{1,2}

А. В. Грязнов ¹, Р.Б. Рыбка ¹, М.С. Скороходов ^{1,2}, И.А. Молошников ¹

¹ Национальный исследовательский центр «Курчатовский институт»

² Национальный исследовательский ядерный университет «МИФИ»

Аннотация. Разработка эффективных интерфейсных систем «человек-машина» для управления робототехническими устройствами при помощи естественного языка является востребованной задачей, в особенности в том случае, когда такое управление осуществляется оператором без специальной подготовки. В работе представлен процесс обработки сложных русскоязычных команд в формализованный RDF формат для управления робототехническим устройством с использованием языковых моделей на основе современной архитектуры трансформер. В данном процессе последовательно задействованы нейросетевые модели для поиска и замены местоимений, восстановления пропущенных глаголов, декомпозиции сложных команд на простые и классификации простых команд. С использованием реализованного в НИЦ «Курчатовский институт» симулятора робототехнической платформы продемонстрирована высокая эффективность процесса обработки сложных русскоязычных команд в составе управляющего интерфейса человека с робототехническим устройством.

Ключевые слова: взаимодействие человека и робота, обработка естественного языка, глубокое обучение.

Введение

Управление в полевых условиях, в том числе неопытными операторами требует создания гибкой и точной системы обработки команд на естественном языке в формализованный формат команд, распознаваемый робототехническим устройством. Представление команды может быть формализовано в виде логического представления или графа, отражающего семантические отношения между сущностями. Популярным формализованным форматом является семантическая графовая модель представления данных RDF [1]. Данный формат оперирует высказываниями вида «субъект» – «предикат» – «объект».

© Сбоев А.Г., Грязнов А.В., Рыбка Р.Б., Скороходов М.С., Молошников И.А., 2023

Множество высказываний образует ориентированный граф с вершинами «субъектами» и «объектами», а ребра отражают «предикаты». Главным отличием формализованного формата от естественного языка является отсутствие неоднозначностей толкования команды. Данное отличие ставит актуальную задачу построения системы обработки русскоязычных команд в формализованный формат, которая включает обработку анафор, свободного порядка слов, неологизмов, омонимии, синонимов и т.д.

Управление робототехническими устройствами на английском языке с применением нейросетевых языковых моделей представлено в ряде работ [2-5]. В работе [2] авторы используют нейросетевые модели с архитектурой LSTM [6] и BERT [7] для преобразования естественного языка в функции языка программирования Python для управления роботизированным манипулятором Dobot Magician [8]. Авторы показывают, что их метод работает лучше, чем обучение нейросетевых моделей прогнозированию действий робототехнической платформы напрямую.

В другой работе [3] используются языковые модели BERT [7] для классификации атрибутов команды и составления последовательности подзадач для управления виртуальным кухонным помощником. Данные модели входят состав модульной системы управления, которая способна управлять робототехнической платформой в отсутствие экспертных траекторий или низкоуровневых инструкций.

Работа [4] предлагает нейросетевой подход с использованием знаний большой языковой модели PaLM [9] для управления роботом от компании Everyday Robots [10]. Авторами проводится оценка полученного подхода на реальном робототехническом устройстве на ряде задач. В результате языковая модель PaLM справляется с двусмысленными и неоднозначными командами естественного языка.

Авторами [5] предложена адаптация на основе языковых моделей GPT-2 [11] и T5 [12] для обработки инструкции на естественном языке при управлении промышленными робототехническими устройствами. Показано, что предварительно обученная языковая модель может быть эффективно настроена для перевода вербальных инструкций в задачи робототехнического устройства лучше, чем методы семантического анализа.

В данной работе представлен метод глубокого обучения на основе языковых моделей с архитектурой трансформер для обработки русскоязычных команд в формализованный графовый формат RDF для управления робототехническим устройством. Данный метод включает модель для поиска и замены местоимений и нейросетевой интерфейс

для обработки сложных русскоязычных команд [13], включающий этапы восстановления пропущенных глаголов, декомпозиции сложных команд на простые и классификации атрибутов простых команд. В разделе 1 представлено описание данных, используемых при обучении моделей. В разделе 2 описаны архитектуры моделей. Раздел 3 описывает полный процесс обработки русскоязычной команды и точности при управлении робототехническим устройством.

1. Набор данных

Синтетический набор данных. Для обучения нейросетевых моделей поиска и замены местоимений, разделения сложных команд на простые и классификации атрибутов простых команд использовался разработанный нами генератор текстовых команд, описанный в работе [13] и модифицированный в этой работе. Данный генератор использует словари с синонимами и инструменты библиотеки `rumorphy2` [14] такие как, согласование слов с числительными и приведение слов в необходимую речевую форму. Модифицированный генератор выполняет следующие функции:

1. Создание множества команд на естественном языке на основе заданных шаблонов.
2. Составление для каждой команды векторного и маркированного представлений.
3. Создание множества сложных команд в виде последовательностей заданных шаблонов.

К простым командам относятся команды на естественном языке, которые состоят из одного типа действия. В общей сложности было составлено 16 шаблонов (см. табл. 1) на различные команды, которые поддерживаются робототехническим устройством. Для классификации атрибутов каждого типа команды генератор составляет векторное представление в зависимости от использованного шаблона и словаря синонимов. Помимо векторного представления генератор составляет маркированное представление, т.е. для каждого слова в сгенерированной команде на естественном языке ставится в соответствие маркер на основе словаря синонимов (см. табл. 2).

Таблица 1

Шаблоны простых команд

№	Описание команды	Примеры генератора
1	Патрулирование	Патрулируй
2	Прекращение выполнения	Стоп
3	Прерывание выполнения	Прервись

Окончание табл. 1

№	Описание команды	Примеры генератора
4	Продолжение выполнения	Продолжай
5	Движение в направлении	Двигайся вперед
6	Поворот в направлении	Поворачивай налево
7	Движение на кол-во метров	Проезжай вперед 2 метра
8	Поворот на кол-во градусов/часов	Поворачивай на 5 часов
9	Взаимодействие с объектом	Подъезжай к дереву
10	Взаимодействие с ближайшим объектом	Осмотри ближайшее дерево
11	Взаимодействие относительно 1 объекта	Осмотри камень, слева от дома
12	Взаимодействие относительно 2 объектов	Поезжай к человеку около дома рядом с деревом
13	Относительно робота	Иди к дереву справа от тебя
14	Направление взгляда	Поезжай к этому человеку
15	Патрулирование по пути	Патрулируй по 2 маршруту
16	Следование за объектом	Следуй за машиной

Таблица 2

Векторное и маркированное представления

Описание атрибута	Вектор	Маркер
Тип действия	0-14	A
Направления движения	0-8	D
Количество метров	0-12	M
Количество градусов	0-24	DS
Количество часов	0-24	HS
Тип объекта 1	0-16	O1
Модификатор ближайшего объекта	0-1	N
Отношение между объектом 1 и объектом 2	0-9	R1
Тип объекта 2	0-14	O2
Отношение между объектом 2 и объектом 3	0-9	R2
Тип объекта 3	0-14	O3
Отношение между объектом 1 и роботом	0-8	S
Модификатор направления взгляда	0-1	G
Вспомогательные слова	–	O

К сложным относятся команды на естественном языке, которые содержат последовательности простых команд. Всего было составлено 4

шаблона (см. табл. 3), которые включают последовательные команды, разделенные ключевыми словами, а также команды, использующие для связи местоимения.

Таблица 3

Шаблон сложных команд

№	Описание команды	Примеры генератора
1	Последовательные 2 команды	Поезжай к дому, потом направляйся к человеку
2	Последовательные 3 команды	Осмотри этот камень, далее подъезжай к дому, а потом найди человека
3	Команды «найди-подъезжай»	Найди дерево. За деревом человек, к которому нужно подъехать
4	Команды, связанные местоимением	Поворачивай к ближайшему дому и осматривай его

Дополнительно для задачи поиска и замены местоимений данные из генератора, в частности шаблон сложных команд с местоимениями, были модифицированы в соответствии с табл. 4.

Таблица 4

Варианты разметки выходных данных для обработки местоимений

№	Входная последовательность	Выходная последовательность
1	Подойди к дому и осмотри его	Команда: подойди к дому и осмотри дом
2	Поворачивай к этому человеку, осматривай его	Цель: человека его

Решение задачи поиска и замены местоимений в командах является частным случаем задачи кореференции (поиск анафор в тексте). Помимо набора данных (см. табл. 4) был подготовлен корпус в соответствии с соревнованием [15] используемый для оценки по метрикам кореференции. В данном соревновании каждому слову в тексте ставится в соответствие позиция слова Offset, количество символов Length и номер связи Chain ID.

Набор данных, полученный в процессе краудсорсинга. Тестирование нейросетевых моделей разделения сложных команд и классификации атрибутов осуществляется на наборе данных, собранном при помощи технологии краудсорсинга. Из синтетического набора данных выбрано 250 образцовых команд, после чего участникам

краудсорсинговой платформы было дано задание перефразировать данные команды (см. табл. 5) разными способами [13].

Таблица 5

Сводка краудсорсинговых тестировочных команд

Тип команды	Кол-во примеров
Команда без атрибутов	51
Движение в направлении	89
Движение к объекту	53
Движение к объекту относительно 1 объекта	130
Движение к объекту относительно 2 объектов	995

В дополнение к генератору команд были проведены сбор и разметка команд на естественном языке для обучения и тестирования классификатора атрибутов простых команд. Целевой группе участников были выданы инструкции, кратко описывающие команды, которые возможно использовать для управления робототехническим устройством. Для каждого вида простых и сложных команд участники составили по 100 примеров. В каждой инструкции был указан набор возможных атрибутов конкретной команды и примеры генератора (см. табл. 6).

Таблица 6

Пример инструкции для участников

Команда	Движение на кол-во метров в направлении
Возможные атрибуты команды	Действие: двигаться Направление: восток, направо, запад, налево Количество метров: 5 метров, 5 м
Шаблон	[действие]+[направление]+[количество метров]
Примеры генератора	Пойди на 2м; Поезжай на 24 м вперед;

Открытые данные. Для обучения модели восстановления пропущенных глаголов был использован набор данных с конференции Dialog-21 [16]. В наборе используются новостные, художественные и технические тексты, а также тексты из социальных сетей. Содержание набора данных представлено в табл. 7, в соответствии с [13]. Дополнительно были использованы ~115000 предложений с автоматической разметкой.

Характеристика открытых данных

Предложения	Обучение	Тестирование	Валидация
С пропусками	5542	680	1382
Всего	16406	2045	4142

2. Методы

Поиск и замена местоимений. Для решения данной задачи использовалась нейросетевая языковая модель `ruT5` с архитектурой трансформер, предназначенная для понимания и генерации текста. Была проведена настройка двух версий модели «base» и «small», отличающиеся количеством параметров. Настройка данных моделей проводилась по примеру из табл. 4.

Другая использованная модель с архитектурой трансформер – это модель на базе `ruBERT`, которая является адаптацией модели [17], основанной на модели кореференции под русский язык [18]. Данная модель взята из библиотеки `Tensorflow` [19] без изменения значений гиперпараметров. Для настройки данной модели использовалось представление данных по примеру следующему примеру:

Текст: [SPL] Иди к дому и осмотри его [SEP] → Связь: [3, 3] [6, 6]

Восстановление пропущенных глаголов. Для решения данной задачи использовался метод [20]. В основе данного метода применяется нейросетевая языковая модель обработки последовательностей текстов. Данная модель выполняет классификацию токенов входного текста по пяти классам: `cV`, `cR1`, `cR2`, `R1`, `R2`. Здесь `cV` - глагол (или предикат), который пропущен в следующих простых предложениях в составе сложного. `cR1` и `cR2` - корреляты из предложения без пропуска, которые синтаксически и семантически схожи с `R1` и `R2` - ремнантами из предложений с пропуском [13].

Разделение сложной команды на простые. Для решения данной задачи использовался классификатор токенов. В качестве классификатора используется нейросетевая языковая модель, которая определяет следующие классы [13]:

1. `O` – не относится ни к одной из команд.
2. `[SEP]` – является частью текущей команды.
3. `[CMD]` – относится к текущей команде.

Классификация атрибутов простой команды. Для решения данной задачи используются 2 нейросетевые языковые модели. Первая модель с топологией трансформер решает задачу классификации токенов. В задаче классификации используются значения векторов и

маркеров из табл. 2. После классификации токенов, из входной строки извлекаются подстроки для каждого из атрибутов.

Вторая модель с топологией трансформер решает задачу соотношения выделенных подстрок атрибутов со словарями этих атрибутов. Для каждого атрибута команды собран словарь, содержащий пары из фразы и значения атрибута. Все фразы словарей кодируются моделью. После этого, выделенные во входной строке атрибуты подаются на вход модели и каждый атрибут также получает один вектор. Этот вектор сравнивается с векторами соответствующего словаря из которого выбирается наиболее близкий вектор по косинусной близости.

Постобработка результатов моделей заключается в выборе максимальной косинусной близости среди всех фраз словаря с данным значением.

3. Эксперимент

Обработка команд на естественном языке включает последовательное выполнение следующих этапов: преобразование голосовой команды, поиск и замена местоимения в команде, разделение сложной команды на простые, восстановление пропущенных глаголов, получение атрибутов команды, составление RDF команды, отправка команды в мир робототехнического устройства.

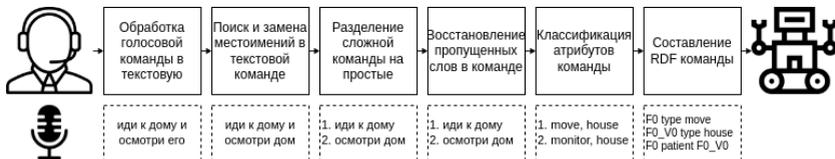


Рис. 1. Процесс обработки русскоязычной команды

Для анализа эффективности представленного решения была проведена оценка точности выполнения команд как с использованием поиска и замены местоимений в сложных командах, так и без данного шага. При оценке точности проверялись атрибуты команды, в основном класс объекта (см. табл. 8).

Таблица 8

Влияние замены местоимений при обработке команд оператора

Команда оператора	С заменой	Без замены
Иди к дому и осмотри его	1 house 2 house	1 house 2 human
Поверни к дереву и подойди к нему	1 tree 2 tree	1 tree 2 gaze

Таким образом, было проверено 1000 команд с местоимениями, используемых при оценке кореференции моделей (см. раздел 1). Данная проверка была проведена для каждой из описанных моделей в разделе 2 Поиск и замена местоимений. Точности правильного определения объекта в команде с местоимением представлены в табл. 9.

Таблица 9

Оценка точности распознавания команд с местоимениями

Процесс обработки	Успешно обработанных команд, %	Среднее время обработки, с
Ru-T5-small, текст команды	32.6	0.232
RuT5-base, текст команды	69.8	0.264
RuT5-small, объект-местоимение	95.4	0.132
RuT5-base, объект-местоимение	98.7	0.117
Без замены местоимений	5	0.044

Заключение

Полученная система на основе нейросетевых языковых моделей позволяет эффективно обрабатывать сложные русскоязычные команды в формализованный RDF формат для управления робототехнической платформой. Данная система работает наиболее точно с моделью для поиска и замены местоимений ruT5-base, которая заменяет местоимение в исходной команде, вместо генерации полного текста команды. Проведенные исследования показывают, что поиск и замена местоимений существенно увеличивает точность интерпретации нейросетевого интерфейса на основе восстановления пропущенных глаголов, декомпозиции сложных команд на простые и классификации атрибутов простых команд.

Список литературы

1. Brickley D. RDF Schema 1.1 / D. Brickley, B. McBride, R.V. Guha // W3C recommendation. – 2014. – Т. 25. – С. 2004.
2. Gubbi S.V. Translating Natural Language Instructions to Computer Programs for Robot Manipulation / S.V. Gubbi, R. Upadrashta, B. Amrutur // arXiv:2012.13695. – 2020.
3. FILM: Following Instructions in Language with Modular Methods / S. Y. Min [и др.] // arXiv:2110.07342. – 2021.
4. Do As I Can and Not As I Say: Grounding Language in Robotic Affordances / M. Ahn [и др.] // arXiv:2204.01691. – 2022.

5. Controlling Industrial Robots with High-Level Verbal Commands / C. Dongkyu [и др.] // ICSR 2021, Social Robotics. – 2021. – С. 216-226.
6. Hochreiter S. Long Short-term Memory / S. Hochreiter, J. Schmidhuber // Neural computation. – 1997. – Т. 9. – С. 1735-1780.
7. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding / J. Devlin [и др.] // arXiv:1810.04805. – 2018.
8. DOBOT [Электронный ресурс]. – Режим доступа: <https://en.dobot.cn/>
9. PaLM: Scaling Language Modeling with Pathways / A. Chowdhery [и др.] // arXiv:2204.02311. – 2022.
10. Everyday Robots [Электронный ресурс]. – Режим доступа: <https://everydayrobots.com/>
11. Language Models are Unsupervised Multitask Learners / A. Radford [и др.] // OpenAI. — 2019.
12. Raffel C. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer / C. Raffel, N. Shazeer, A. Roberts // arXiv:1910.10683. — 2019.
13. Нейросетевой интерфейс конвертации сложных русскоязычных текстовых команд в формализованный графовый вид для управления робототехническими устройствами / А.Г. Сбоев [и др.] // Вестник Национального исследовательского ядерного университета МИФИ. – 2022. – Т.11 № 2. – С. 153-163.
14. Korobov M. Morphological Analyzer and Generator for Russian and Ukrainian Languages / M. Korobov // Analysis of Images, Social Networks and Texts. — 2015. — С. 320-332.
15. Ru-eval-2019: Evaluating anaphora and coreference resolution for Russian / Е.А. Budnikov [и др.] // Dialogue Evaluation. – 2019.
16. Agrr-2019: Automatic gapping resolution for Russian / I.M. Smurov [и др.] // Computational Linguistics and Intellectual Technologies. – 2019. – С. 561-575.
17. BERT for Coreference Resolution: Baselines and Analysis / M. Joshi [и др.] // Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing. — 2019. — С. 5803-5808.
18. Sboev A. Deep Neural Networks Ensemble with Word Vector Representation Models to Resolve Coreference Resolution in Russian / A. Sboev, R. Rybka, A. Gryaznov // Advanced Technologies in Robotics and Intelligent Systems. — 2020. — С. 34-35.
19. Tensorflow [Электронный ресурс]. – Режим доступа: <https://www.tensorflow.org/>