УДК 551.46.077:629.584
EDN: VQOQQZ
DOI: 10.37102/1992-4429_2025_53_03_06

СРАВНИТЕЛЬНЫЙ АНАЛИЗ НЕЙРОННЫХ СЕТЕЙ-ТРАНСФОРМЕРОВ ПРИ РЕШЕНИИ ЗАДАЧИ ГЕНЕРАЦИИ МИССИЙ АНПА

А.С. Пугачев, А.И. Боровик

В статье рассматривается задача автоматизации составления программ-миссий для автономных необитаемых подводных аппаратов (АНПА) с использованием нейронных сетей. Основное внимание уделено выбору оптимальной модели для преобразования команд с естественного языка в код на специализированном языке программирования подводных исследований – ЯППИ. Рассмотрены модели LLaMA 3.1 8B-instruct, Gemma 2 9B IT, Mistral 7B-instruct, Qwen 7B-instruct, Phi-4 3.8B, проведено их сравнение по критериям точности генерации, скорости выполнения, устойчивости к ошибкам во входных данных и ресурсоемкости. Результаты проведенного сравнительного анализа показали, что модели LLaMA 3.1 8B-instruct и Phi-4 3.8B лучше других решают поставленную задачу. При этом LLaMA 3.1 обладает более высокой скоростью обработки и способностью корректно интерпретировать команды, что делает ее предпочтительной для использования в разрабатываемой интеллектуальной системе поддержки деятельности операторов АНПА. Для повышения точности и адаптивности модели предложен подход к дообучению нейросети на расширенной обучающей выборке, созданной с использованием более мощных языковых моделей.

Ключевые слова: АНПА, нейронные сети, миссия, язык программирования миссий АНПА, ЯППИ.

Введение

Автономные необитаемые подводные аппараты (АНПА) в настоящее время активно применяются для выполнения широкого спектра морских работ, таких как поиск затонувших объектов, инспекция подводных трубопроводов и телекоммуникационных кабелей, разведка морских ресурсов, экологический мониторинг, сбор данных для морских научных исследований и многих других. Благодаря способности осуществлять продолжительные миссии на значительных глубинах АНПА позволяют получать доступ к труднодоступным подводным зонам, сокращая необходимость в использовании дорогостоящих и ресурсоёмких средств, таких как пилотируемые батискафы или группы водолазов [1]. Однако эффективное применение АНПА требует наличия квалифицированной команды операторов, в задачу которых входит формирование детального плана работ в конкретной акватории, составление программ-миссий для системы управления робота, организация работ по спуску аппарата на воду и его подъему, удаленное сопровождение АНПА в ходе выполнения задания и камеральная обработка полученных данных. На практике одной из ключевых проблем применения АНПА остаётся дефицит специалистов, обладающих достаточным опытом работы с конкретными типами подводных роботов и готовых к участию в морских экспедициях [2]. В то же время в современном мире наблюдается стремительнее развитие нейронных сетей – все больше и больше сфер человеческой деятельности автоматизируются с их использованием – от создания рекламных буклетов, до анализа медицинских снимков и управления сложными системами. В области применения АНПА нейросети также могут стать инструментом, облегчающим труд операторов.

Одной из немаловажных задач, которую необходимо решить для автоматизации проведения подводных исследований с использованием АНПА, является задача составления миссий аппарата на основе запросов, сформулированных на естественном языке. Такие запросы могут генерироваться оператором или руководителем группы применения АНПА – че-

ловеком, знакомым с характером проводимых работ, но не знающим специфики конкретного робота или языков программирования миссий. Нейросети могут значительно ускорить и упростить процесс создания программ-миссий, автоматически генерируя код на основе информации, собранной в ходе диалога с оператором. Их применение позволит сократить временные затраты на написание и проверку миссий, облегчить обмен опытом между операторами и снизить когнитивную нагрузку на них, тем самым повышая эффективность проведения работ с АНПА.

Цель данной статьи заключается в подборе оптимальной нейронной сети для генерации кода миссии АНПА на основе запроса на естественном языке. В качестве языка запросов будет использоваться русский, в качестве языка миссий аппарата – ЯППИ (язык программирования подводных исследований) [3]. ЯППИ в качестве целевого языка выбран в связи с тем, что сгенерированный на нем код, используя разработанный ранее интерпретатор, может быть преобразован в целевой код миссии для любых АНПА производства ИПМТ ДВО РАН. Подбираемая нейросеть будет использоваться в создаваемой системе интеллектуальной поддержки деятельности операторов в качестве промежуточного звена между пользователем и интерпретатором ЯППИ. Ее главная задача – корректно интерпретировать намерения пользователя, сформулированные на естественном языке, и преобразовывать их в код на ЯППИ.

1. Требования к системе

Алгоритм получения кода миссии с использованием нейросети выглядит следующим образом:

1) пользователь отправляет запрос на создание миссии (либо методом набора текста в поле ввода интерфейса системы, либо путем произнесения команды в микрофон с ее последующим распознаванием через систему «речь-в-текст»);

2) нейросеть анализирует запрос и генерирует соответствующий ему код на ЯППИ;

3) сгенерированный код передается интерпретатору ЯППИ для дальнейшей обработки и трансляции на «нативный» язык описания миссий аппарата.

Если в процессе выполнения трансляции возникает ошибка, код возвращается нейросети для исправления. В случае невозможности автоматического устранения проблемы нейросеть должна уточнить у пользователя необходимые данные и повторить генерацию миссии (рис. 1).

Многие требования к нейросети напрямую следуют из планируемого характера ее использования в качестве составного элемента интеллектуальной системы поддержки деятельности операторов. Система должна работать на относительно низкопроизводительном мобильном компьютере, не имеющем доступа в сеть Интернет (что актуально при проведении любых морских работ вдалеке от крупных городов). Использовать систему будет оператор – человек, знакомый с терминологией морских работ и основными аспектами применения подводной робототехники, который может использовать в запросах специфические термины и «сленг». Код, генерируемый нейросетью на ЯППИ, сначала будет проходить через анализаторы и модули (уточнения параметров, низкоуровневых команд) ЯППИ – поэтому нейросеть сразу будет получать запрос на исправление или дополнение кода миссии, если совершит при его генерации ошибку. С учетом этого наиболее важные требования к выбираемой нейросети можно сформулировать следующим образом:

- *точность генерации* – модель должна правильно интерпретировать входной текст и генерировать синтаксически корректные команды на ЯППИ;
- *скорость выполнения* – модель должна максимально быстро генерировать миссию АНПА в условиях работы в реальном времени;
- *робастность* – модель должна быть устойчивой к шуму во входных данных, такому как «сленговые» выражения, опечатки, неоднозначности или неточные формулировки;
- *диалоговый режим* – модель должна уточнять у оператора его намерения, если исходных дан-

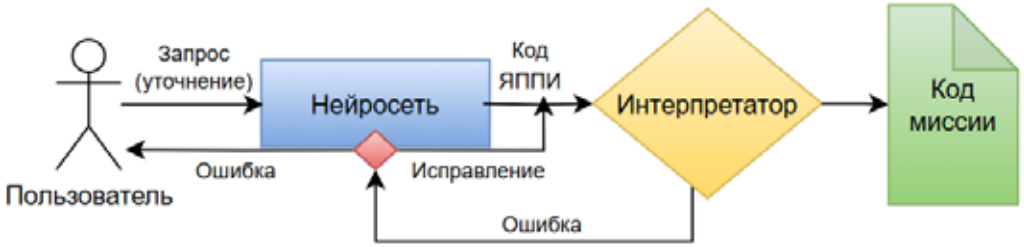

Рис. 1. Алгоритм работы нейросети для генерации корректного кода миссии АНПА

ных в запросе недостаточно для генерации корректной миссии; в то же время модель должна уметь корректировать код, если интерпретатор сообщает о найденных в нем ошибках;

- *эффективность использования ресурсов* – модель должна быть достаточно легковесной для развертывания локально на мобильном компьютере.

2. Нейросети, участвующие в тестировании

Для решения задачи генерации кода миссии по запросу на естественном языке наиболее подходящими являются глубокие нейронные сети с архитектурой «трансформер» [4]. Подобные модели широко применяются в задачах обработки естественного языка (Natural Language Processing, NLP) благодаря своей способности эффективно работать с длинными последовательностями и контекстуальной зависимостью [5, 6]. Для тестирования были выбраны языковые модели с открытой лицензией, предварительно обученные на огромных объемах данных [7]. Открытая лицензия позволяет свободно использовать данные нейросети в разрабатываемой системе, а проведенное ранее предварительное обучение позволяет заменить полноценное обучение (естественному языку) на дообучение специфике применения АНПА и языку ЯППИ.

В данной работе рассмотрены следующие языковые модели:

- LLaMA 3.1 8B-instruct (разработчик Meta);
- Gemma 2 9B IT (разработчик Google DeepMind);
- Mistral 7B-instruct (разработчик Mistral AI);
- Phi-4 3.8B (разработчик Microsoft);
- Qwen 7B-instruct (разработчик Alibaba Group).

3. Обучающая выборка

Все сравниваемые нейросети являются «предобученными» и подходят для решения широкого спектра задач, включая обработку естественного языка и генерацию кода для известных языков программирования, таких как Python или JavaScript. Однако их применение для обработки специфических запросов, описывающих применение АНПА для проведения подводных работ, и генерации кода на специализированных языках, таких как ЯППИ, требует проведения «дообучения» [8, 9]. Для проведения «дообучения» была сформирована обучающая выборка, состоящая из 200 пар, содержащих описание миссии АНПА на русском языке и ее код на ЯППИ. Для составления выборки использовались как реальные миссии, взятые из экспедиционных протоколов ИПМТ ДВО РАН, так и синтезированные миссии, разработанные при участии действующих операторов. Фрагмент выборки приведен на рис. 2.

```
{
"text": "Произвести обследование прямоугольного участка акватории с
  координатами левой нижней точки: 131.911296 восточной долготы, 43.107656
  северной широты. Длина – 800 м, ширина – 500 м. Использовать гидролокатор
  бокового обзора. Траектория движения – вертикальный меандр, интервал между
  галсами – 80 м, высота – 7 м, скорость – 1 м/с. Средняя глубина района – 100
  м.",
"labels": "миссия(глубина_места(100)) обследование_фигуры(фигура(прямоугольник
  ), координаты(131.911296*ВД, 43.107656*СШ), длина(800), ширина(500), межгалс
  (80), высота(7), прибор(гбо), траектория(меандр, вертикально), скорость(1))"
},
{
"text": "Обследуй участок трубопровода змейкой по точкам [132.101234, 43
  .209876], [132.104567, 43.209765], [132.107890, 43.209654], используя гбо на
  высота 8м, со скоростью 0.6м/с.",
"labels": "миссия() обследование_линии(траектория(ломанная), координаты([132
  .101234*ВД, 43.209876*СШ], [132.104567*ВД, 43.209765*СШ], [132.107890*ВД, 43
  .209654*СШ]), прибор(гбо), скорость(0.6), высота(8))"
},
{
"text": "Обследуй точку 138.9012 ВД, 50.1234 СШ по спирали с фотокамерой.
  Параметры: высота 7м, скорость 0.8м/с. Глубина местности около 12м.",
"labels": "миссия(глубина_места(12)) обследование_точки(траектория(спираль),
  координаты(138.9012*ВД, 50.1234*СШ), высота(7), скорость(0.8), прибор
  (фотокамера))"
}
```

Рис. 2. Фрагмент обучающей выборки для дообучения нейросети

4. Тестовая выборка

Для выполнения тестирования нейросетей была составлена тестовая выборка, содержащая запросы, обработка которых позволяет оценить точность, скорость, робастность нейросетей, а также их способность поддерживать диалог с пользователем. Фрагмент тестовой выборки приведен на рис. 3.

Для оценки точности использовались два вида запросов: похожие на запросы из обучающей выборки и подразумевающие формирование сложносоставных миссий, являющихся комбинацией нескольких миссий обучающей выборки.

Для оценки робастности использовались запросы, содержащие опечатки и «сленговые» слова.

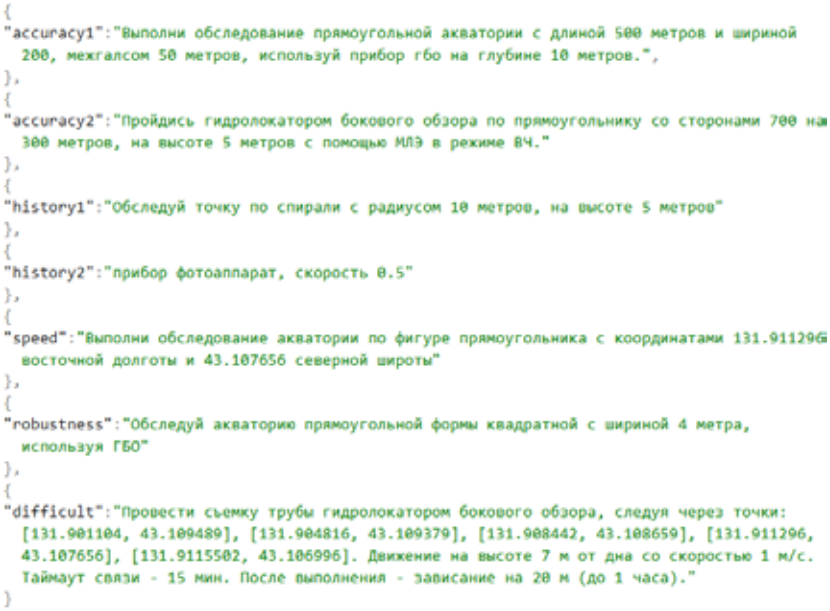

Рис. 3. Фрагмент тестовой выборки

Для оценки качества поддержания диалога использовались запросы, содержащие неполные данные, а также запросы, содержащие цепочку уточнений и правок.

Скорость модели оценивалась путем определения скорости формирования первого символа и скорости генерации последующих токенов. Эффективность использования ресурсов оценивалась по потреблению нейросетью процессорного времени, оперативной и постоянной памяти в ходе выполнения тестирования.

5. Критерии оценивания

Оценивание нейросетей проводилось с использованием системы баллов, где по каждому критерию ей присваивалась оценка от 1 до 5 (больше – лучше). При этом вес каждого критерия определялся его коэффициентом значимости, отражающим влияние критерия на общую эффективность системы.

Наибольший вес в системе оценивания имели точность генерации кода и точность генерации кода для сложносоставных миссий. Эти параметры являются ключевыми, так как именно они отвечают за правильность создания итогового программного кода на ЯППИ. Ошибки, допущенные на данном этапе, могут привести к некорректной работе всей системы.

Следующими по значимости критериями являлись скорость выполнения и робастность. Модель должна не только генерировать корректный код, но и делать это в соответствии с заданными требованиями, избегая лишних задержек. При этом важно минимизировать время ожидания пользователя, чтобы система оставалась удобной для практического применения.

Итоговая оценка модели рассчитывается по формуле:

$$y = \sum_i^n k_i c_i, \quad (1)$$

где

k_i – коэффициент значимости критерия;

c_i – оценка работы нейросети по конкретному критерию;

y – итоговый результат нейронной сети.

Коэффициенты значимости k для каждого критерия приведены в табл. 1.

Таблица 1. Коэффициенты значимости критериев оценивания

Критерий	Коэффициент значимости
Точность	0,25
Точность генерации кода для составных миссий	0,2
Скорость выполнения	0,15
Робастность	0,15
Поддержка диалогов	0,1
Ресурсы	0,1
Объем модели	0,05

6. Аппаратное обеспечение

Дообучение моделей и их тестирование проводились на ноутбуке с параметрами, представленными в табл. 2. Характеристики данного компьютера в целом соответствуют параметрам техники, применяемой в экспедициях для планирования миссий и отслеживания работы АНПА.

Таблица 2. Характеристики ноутбука, использованного во время проведения тестирования

Процессор	Intel Core i7-11800H, частота 2.30 GHz, 8 ядер
Оперативная память	16 ГБ, DDR4
Видеопамять	8ГБ, 1560 МГц

7. Описание хода тестирования

С целью обеспечения корректного сравнительного анализа языковых моделей использовался единый набор входных данных, последовательно

подаваемых на каждую из тестируемых нейросетей. Для оптимизации ресурсов применялись 4-битное квантование, сокращающее объем памяти, и адаптация LoRA, позволяющая обучать модель с минимальным изменением параметров [10, 11]. Ниже представлено описание основных аспектов проведенного тестирования по каждой из оцениваемых нейросетей.

7.1. LLaMA 3.1 8B-instruct

LLaMA 3.1 8B-instruct демонстрирует высокую точность в задачах генерации текста и кода благодаря своей архитектуре, которая оптимизирована для работы с инструкциями и сложными запросами. Модель способна корректно интерпретировать текстовые команды, учитывая контекст и семантику.

Во всех примерах LLaMA 3.1 8B-instruct корректно интерпретировала смысловые нюансы и контекст, генерируя решения, которые соответствуют ожиданиям пользователя.

После генерации ответа нейросеть корректно обрабатывала просьбы о расширении и уточнении запроса, сохраняя изначальную связность и целостность контента (рис. 4).

LLaMA 3.1 устойчива к шумам и ошибкам в командах. Модель способна корректно интерпретировать команды, даже если они содержат незначительные ошибки. На рис. 5 представлен сгенерированный код для сложносоставной миссии, реализация которой требует больше двух строк кода на ЯППИ.

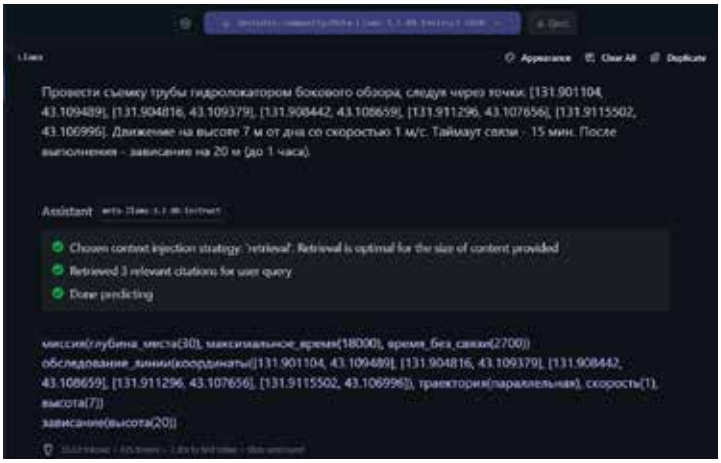

Рис. 5. Точность генерации сложносоставной миссии Llama 3.1 8B-Instruct

Как показало тестирование, LLaMA неплохо справляется с формированием кода для сложной миссии, при этом иногда «придумывая» значения параметров, которые не были заданы (см. на рис. 5 – «*глубина_места*»). Такое поведение не всегда стоит расценивать как благоприятное, поскольку в случае ненамеренного пропуска параметра модуль верификации в ЯППИ не запросит его уточнение.

Модель обеспечивает быструю обработку команд благодаря оптимизированным алгоритмам генерации ответа. LLaMA 3.1 использует эффективные механизмы параллельной обработки данных, что позволяет ей быстро генерировать код. Однако скорость выполнения может снижаться при работе на устройствах с ограниченными вычислительными ресурсами. При запросе на создании простой миссии на ЯППИ, LLaMA 3.1 8B-instruct пока-

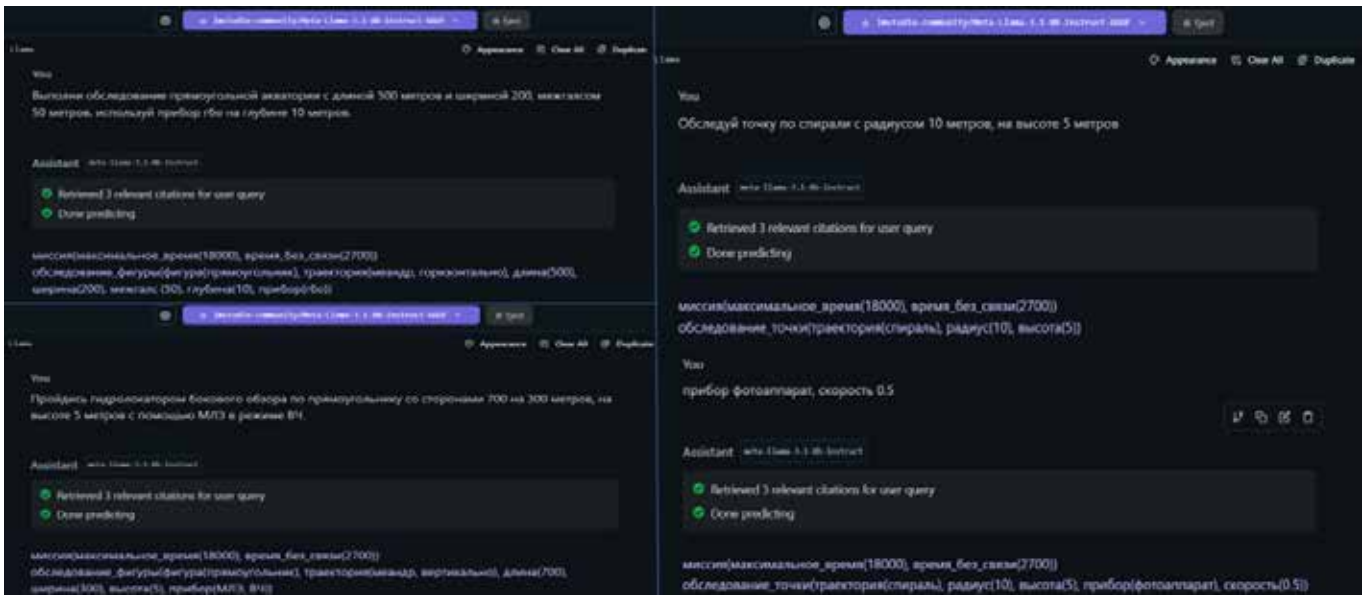

Рис. 4. Точность генерации миссии Llama 3.1 8B-Instruct

зывала высокие значения по генерации токенов – 56.08 токенов в секунду и 1.08 секунды на генерацию первого. Среднее время генерации кода у LLaMA 3.1 составляет примерно 2.52 секунды (рис. 6).

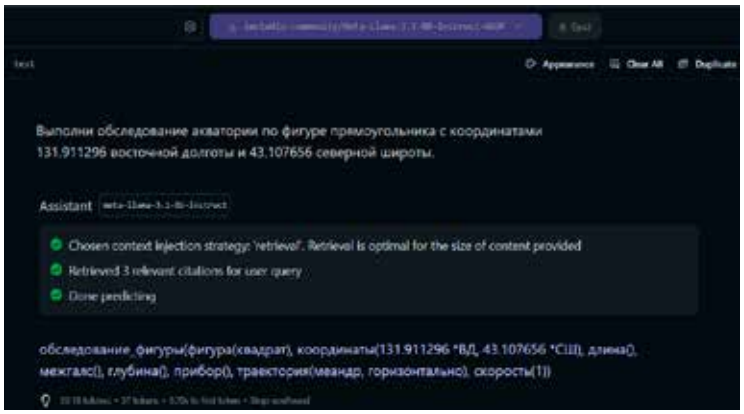

Рис. 6. Скорость генерации кода Llama 3.1 8B-Instruct

Основным преимуществом LLaMA 3.1 8B-Instruct является небольшое потребление вычислительных ресурсов. В отличие от более объемных версий из семейства LLaMA, таких как LLaMA 3.1 70B и LLaMA 3.1. 40B, данная версия модели специально разрабатывалась для маломощных устройств, что, в свою очередь, подходит для работы в системе поддержки деятельности операторов.

7.2. Gemma 2 9B IT

Gemma 2 9B IT показывает высокие результаты в задачах генерации кода благодаря большому объему обучающих данных и специализированным настройкам.

В большинстве примеров Gemma 2 демонстрирует высокую степень понимания намерений пользователя и ожидаемый вывод.

После генерации ответа Gemma 2 корректно обрабатывает просьбы о расширении запроса, сохраняя изначальную связность контента.

Масштаб данных, на которых обучалась Gemma 2, позволял ей хорошо справляться с шумами и ошибками (рис. 7).

Во время генерации сложносоставных миссий (см. рис. 8) Gemma допускала ряд неточностей, связанных с правильностью интерпретации входных значений. Например, модель указывала высоту движения в параметре «*глубина_места*», а также добавляла единицы измерений для параметров, что нарушает синтаксис языка (во всяком случае, для текущей версии ЯППИ).

Скорость выполнения Gemma 2 ниже, чем у LLaMA 3.1, что связано с большей сложностью модели. Однако модель все же обеспечивает приемлемую скорость обработки команд, особенно при использовании на мощных вычислительных устройствах. При том же запросе на создании простой миссии на ЯППИ, что и у LLaMA 3.1 8B-instruct, Gemma 2 9B IT показывала следующие значения – 33.45 токенов в секунду и 2.36 секунды на генерацию первого токена. Общее время на генерацию кода у Gemma 2 9B составляет примерно 4.6 секунды (рис. 9).

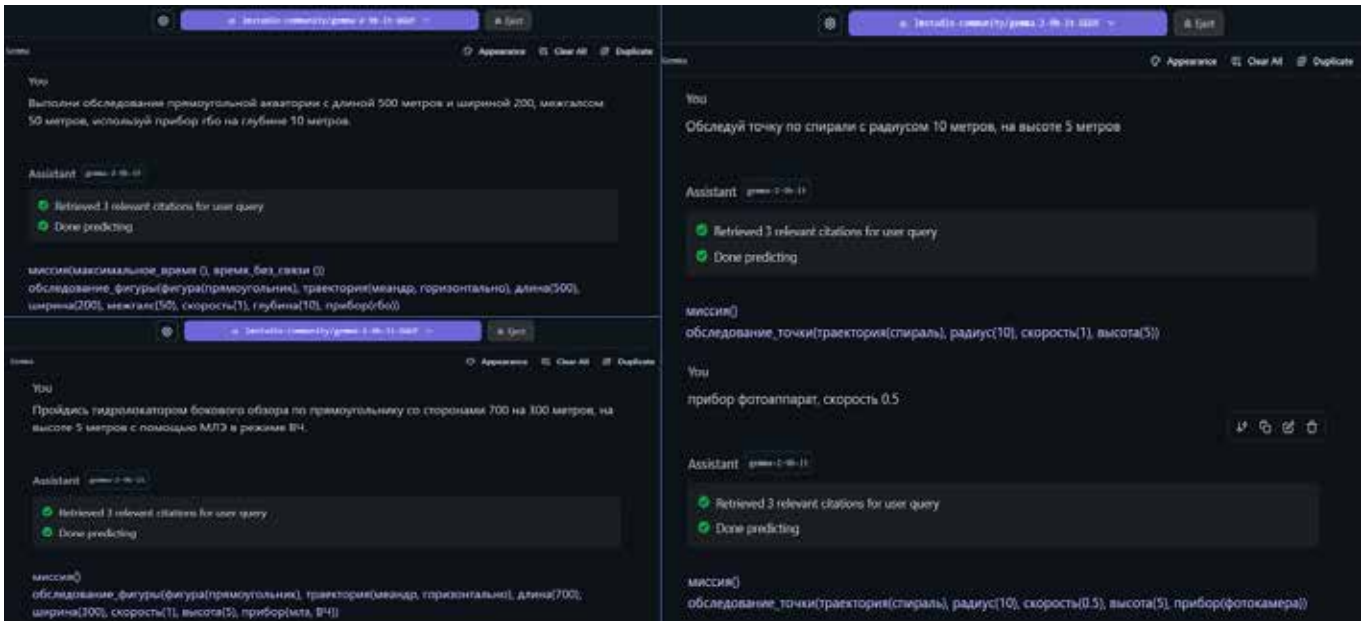

Рис. 7. Точность генерации кода Gemma 2 9B IT

Стоит отметить, что данная модель требует значительных вычислительных ресурсов, что ограничивает ее применение на устройствах с ограниченной мощностью. Этот факт делает Gemma 2 менее подходящей для маломощных устройств, но, в свою очередь, модель является хорошим вариантом на производительных платформах.

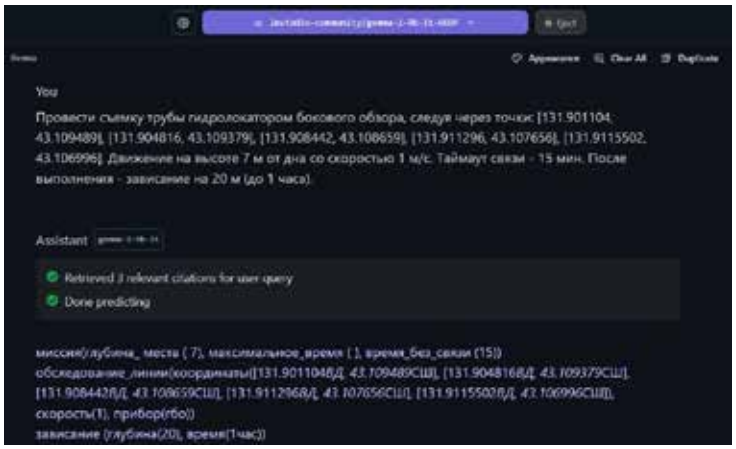

Рис. 8. Точность генерации кода сложносоставной миссии Gemma 2 9B IT

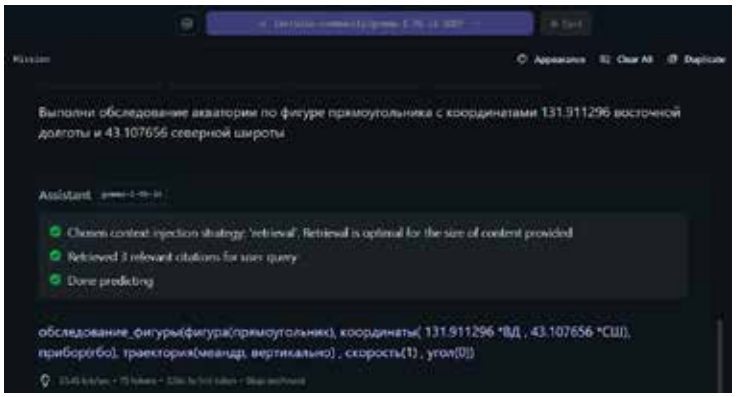

Рис. 9. Скорость генерации кода Gemma 2 9B IT

7.3. Mistral 7B-instruct

Mistral 7B-Instruct демонстрирует высокую точность генерации кода благодаря архитектурным оптимизациям, направленным на работу с инструкциями. При тестировании модели на тренировочных данных система генерировала корректные ответы, сопоставимые с результатами, полученными у моделей LLaMA и Gemma. При обработке запросов, содержащих формулировки, близкие к живой речи и сленг, Mistral 7B-Instruct генерирует корректный код на ЯППИ (рис. 10).

Несмотря на высокий показатель точности при генерации кода, модель демонстрирует слабые стороны при работе с задачами, содержащими «шумы». Это подчёркивает необходимость дополнительной адаптации модели к специфическим требованиям задач.

На рис. 11 видно, что Mistral, как и Gemma, допускала схожие ошибки в интерпретации значений для конкретных параметров. Более того, данная модель не смогла сгенерировать вторую часть кода, связанную с необходимостью зависания аппарата на определенной высоте.

Mistral обеспечивает быструю обработку команд, сравнимую с LLaMA 3.1. Это дости-

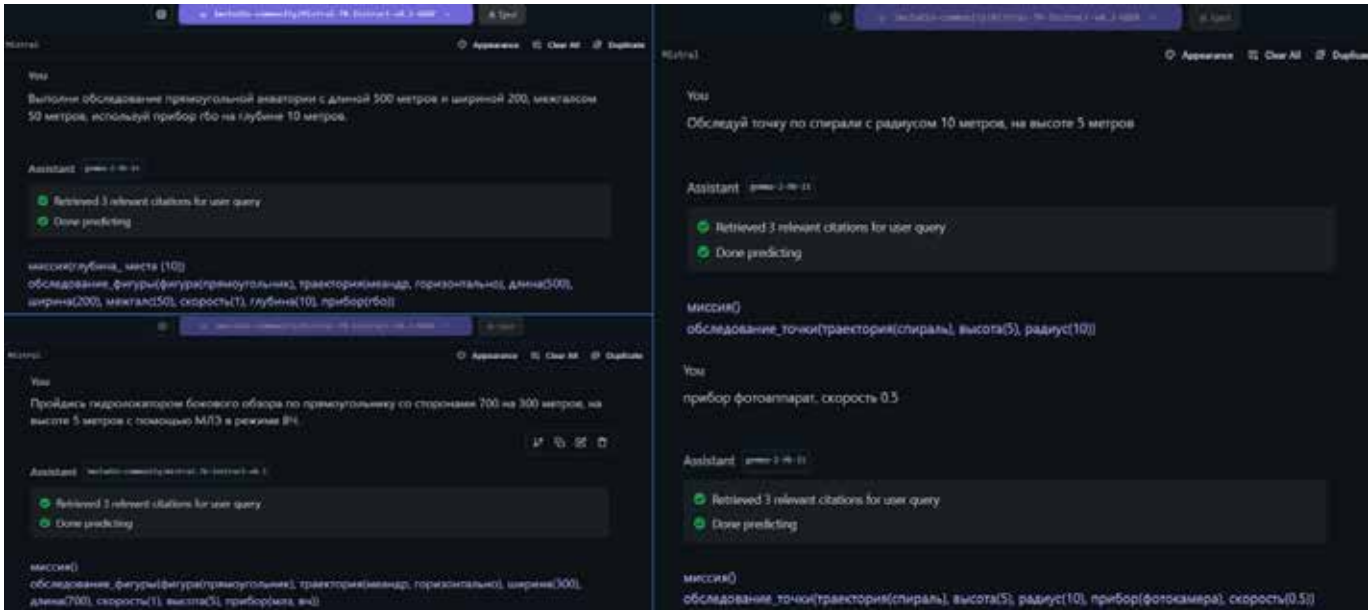

Рис. 10. Точность генерации кода Mistral 7B-instruct

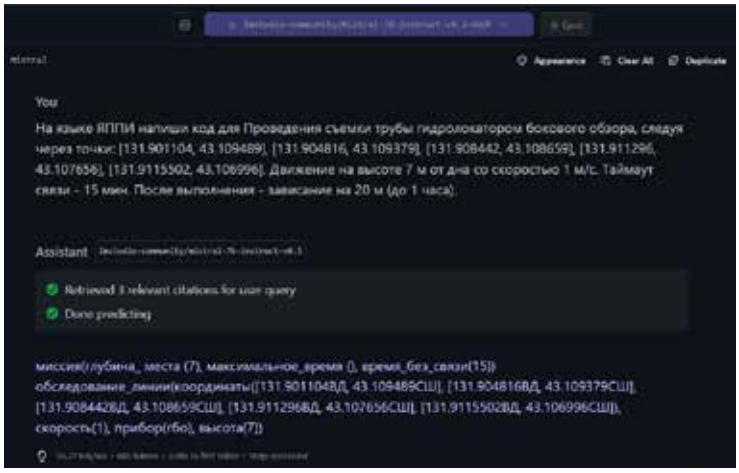

Рис. 11. Точность генерации кода сложносоставной миссии Mistral 7B-instruct

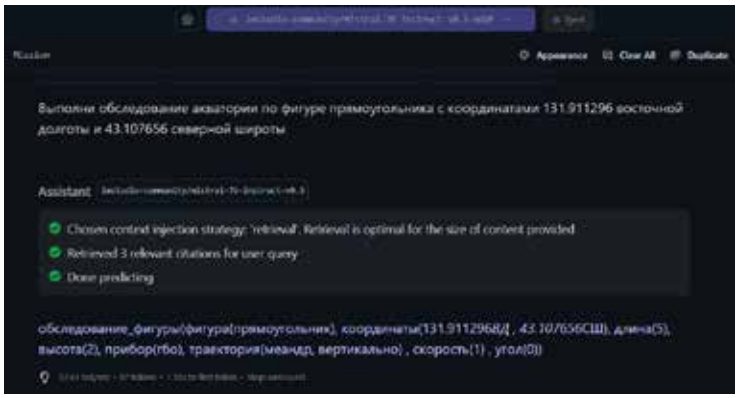

Рис. 12. Скорость генерация кода Mistral 7B-Instruct

гается за счет оптимизированных алгоритмов вывода и эффективного использования вычислительных ресурсов. При генерации миссии на ЯППИ Mistral 7B-instruct показывала следующие значения – 57.61 токенов в секунду и 1.55 секунды на генерацию первого токена. Исходя из данных, представленных на рис. 15, можно сделать заключение, что общее время на генерацию кода у Mistral заняло примерно 3.23 секунды (рис. 12).

Стоит отметить, что модель Mistral требует меньше ресурсов, чем Gemma 2, что делает ее более подходящей для устройств с ограниченной мощностью.

7.4. Qwen 7B-instruct

Qwen 7B-Instruct обучена на большом объеме данных, включая тексты на множестве языков. Модель использует архитектуру трансформер с улучшенными механизмами внимания, позволяющие ей увеличить способность к контекстному пониманию и генерации последовательностей для ответа пользователю.

Qwen 7B-Instruct демонстрировала высокую точность в задачах, близких к её обучающим данным. Для узкоспециализированных языков программирования точность снижается. Модель может генерировать синтаксически правильный код, но допускать семан-

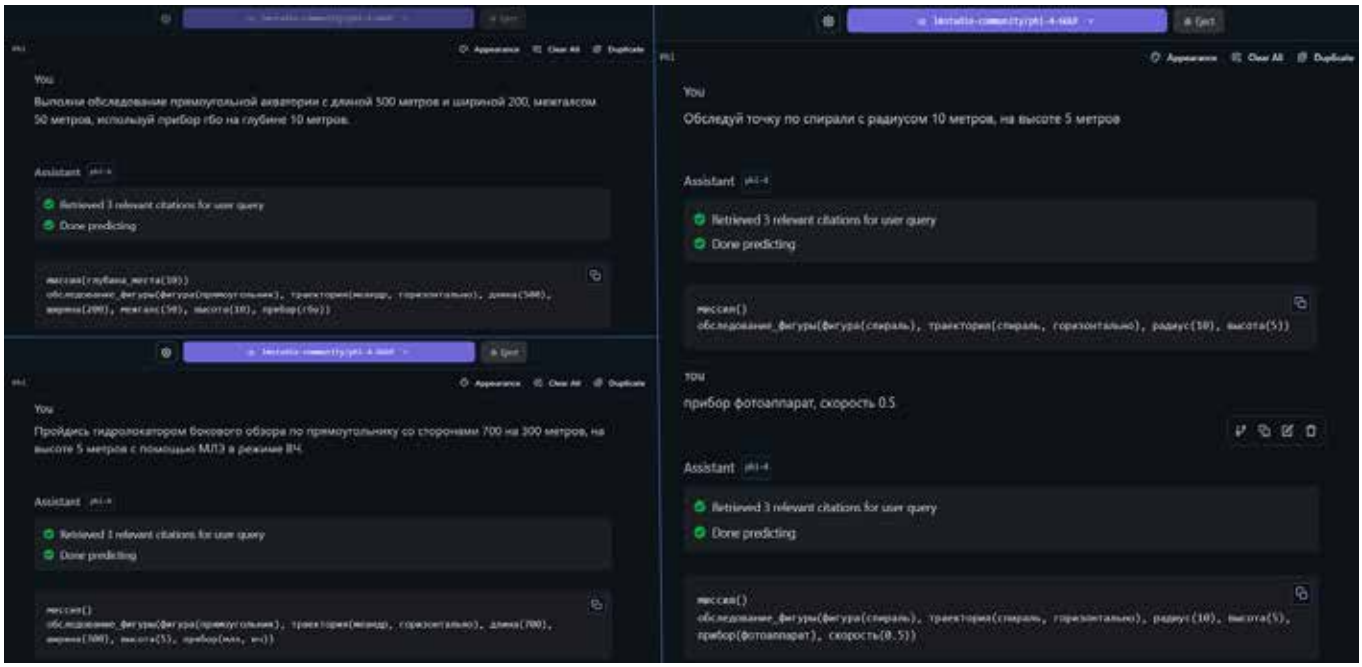

Рис. 13. Точность генерации кода Phi-4 3.8B

тические ошибки. Точность повышается при дообучении на большом объеме данных.

Скорость генерации миссии является довольно медленной, сравнимой по величине с Gemma 2, поскольку Qwen 7B-Instruct также является объемной моделью.

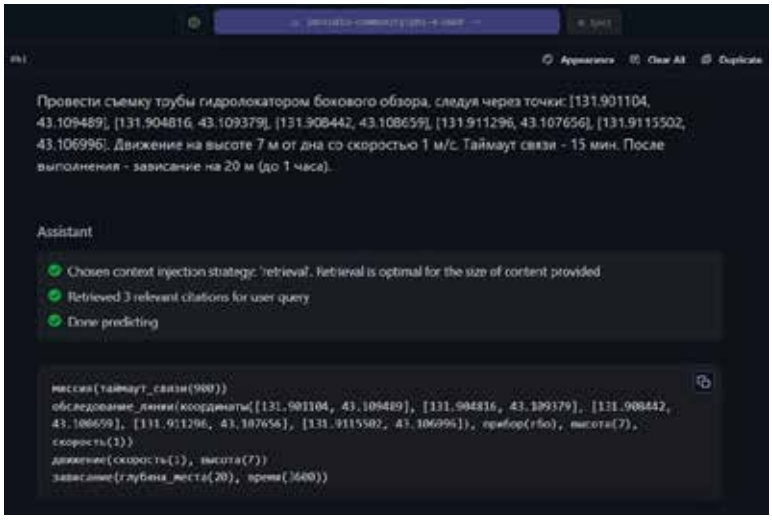

Рис. 14. Точность генерации кода сложносоставной миссии Phi-4 3.8B

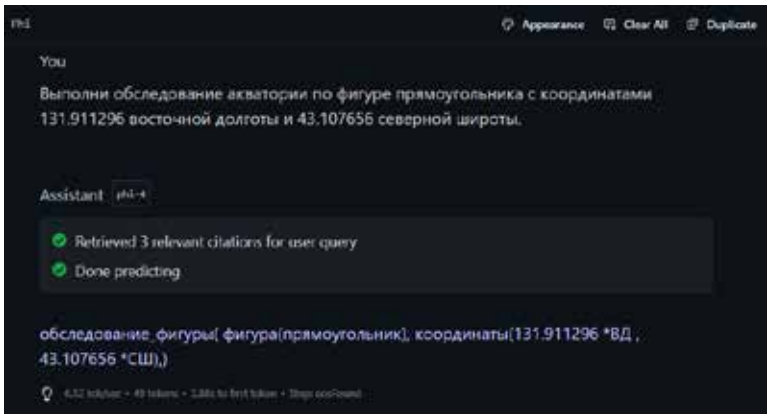

Рис. 15. Скорость генерации кода Phi-4 3.8B

Стоит отметить, что Qwen 7B-Instruct является самой ресурсоёмкой моделью среди рассматриваемых: она требует больше вычислительных мощностей, что делает её проблемной для запуска на устройствах операторов.

7.5. Phi-4 3.8B

Архитектура Phi-4 3.8B оптимизирована для малых размеров без значительной потери точности благодаря улучшенным методам компрессии. Обладая самым низким показателем по потреблению ресурсов, Phi-4 3.8B весьма успешно справляется с задачами по генерации кода, при этом демонстрируя сравнимую точность. Хотя в случае с поддержанием историчности модель может допускать проблемы, связанные с синтаксисом ЯППИ.

На рис. 14 Phi-4 безошибочно сгенерировала код на ЯППИ для сложной миссии.

При миссии на ЯППИ Phi-4 3.8B показывала следующие значения – 6.52 токенов в секунду и 2.88 секунды на генерацию первого токена. Исходя из данных, представленных на рис. 10, можно сделать заключение, что общее время на генерацию кода у Phi-4 3.8B заняло примерно 10.24 секунды (рис. 11).

8. Сравнительный анализ

По результатам проведенного анализа была составлена сравнительная таблица, содержащая итоговые баллы каждой модели,

Таблица 3. Оценки моделей

Показатель	LLaMA 3.1 8B-instruct		Gemma 2 9B IT		Mistral 7B-Instruct		Qwen 7B-instruct		Phi-4 3.8B	
	Баллы	Оценка	Баллы	Оценка	Баллы	Оценка	Баллы	Оценка	Баллы	Оценка
Точность генерации	4,35	1,0875	4,5	1,125	4,25	1,0625	4,5	1,125	4	1
Точность генерации составных миссий	4,5	0,9	4	0,8	3,5	0,7	3,75	0,75	4,7	0,94
Скорость выполнения	4,25	0,6375	3	0,45	4,6	0,69	3,5	0,525	4	0,6
Робастность	4	0,6	2,5	0,375	2,5	0,375	3	0,45	3,5	0,525
Поддержка диалога	5	0,5	5	0,5	5	0,5	5	0,5	5	0,5
Ресурсы	4	0,4	3	0,3	4,5	0,45	1,5	0,15	4,7	0,47
Объем модели	4	0,2	4,5	0,225	3,8	0,19	2	0,1	5	0,25
Итоговый результат ()	4,325		3,775		3,9675		3,6		4,285	

а также оценки по каждому параметру, рассчитанные по формуле (1).

Как видно из проведенного сравнения, наиболее подходящими для поставленной задачи являются модели Phi-4 3.8B и LLaMa 8B-instruct. Обе модели демонстрируют высокие показатели в решении данной задачи. Однако Phi-4 3.8B немного уступает LLaMA 3.1 8B-instruct по точности и способности корректно интерпретировать команды с неточными формулировками. Кроме того, LLaMA-3 обеспечивает более высокую скорость обработки команд.

Таким образом, несмотря на то что обе модели подходят для решения поставленной задачи, LLaMA-3.1 8B-instruct является более предпочтительной для использования в разрабатываемой системе поддержки деятельности оператора.

9. Результаты

По итогам проведенного анализа установлено, что модели LLaMA 3.1 8B-instruct и Phi-4 3.8B демонстрируют наилучшие показатели при генерации программ-миссий на ЯППИ на основе запросов на естественном языке. LLaMA 3.1 выделяется высокой скоростью работы и способностью корректно интерпретировать команды, что делает ее предпочтительной для решения задачи автоматической генерации миссий АНПА.

10. Будущие работы

Подобранная языковая модель является одним из ключевых компонентов разрабатываемой интеллектуальной системы поддержки деятельности операторов АНПА.

Для проведения дообучения модели предлагается расширить текущий объем обучающей выборки за счет использования более мощных языковых моделей (DeepSeek, ChatGPT4, Qwen3). Начальный набор данных будет обработан нейросетью с целью генерации вариаций пользовательских запросов на естественном языке с неизменными ожидаемыми результатами. Такой подход позволит автоматическим путем увеличить объем и разнообразие обучающих данных.

Всего планируется сгенерировать около 3000 тренировочных миссий. Исходя из материалов, представленных в статьях [12, 13], можно предположить, что данного размера обучающей выборки будет достаточно для получения нейросети, способной эффективно выполнять поставленную задачу.

СПИСОК ИСТОЧНИКОВ

1. Агеев М.Д., Касаткин Б.А., Киселев Л.В., Молоков Ю.Г., Никифоров В.В., Рылов Н.И. Автоматические подводные аппараты. Ленинград: Судостроение, 1981. 224 с.

2. Инзарцев А.В., Киселев Л.В., Костенко В.В., Матвиенко Ю.В., Павин А.М., Щербатюк А.Ф. Подводные робототехнические комплексы: системы технологии применения. Владивосток: ДВО РАН ИПМТ, 2018. 368 с.

3. Пугачев, А.С., Боровик А.И. ЯППИ – универсальный язык программирования миссий АНПА // Подводные исследования и робототехника. 2024. № 3(49). С. 26–37. DOI 10.37102/1992-4429_2024_49_03_03. – EDN GVSAMY.

4. Vaswani A., Shazeer N., Parmar N., Uszkoreit J., Jones L., Gomez A. N., Kaiser Ł., Polosukhin I. Attention is All you Need // Advances in Neural Information Processing Systems 30 / I. Guyon, U. v. Luxburg, S. Bengio, H. Wallach, R. Fergus, S.V.N. Vishwanathan, R. Garnett – 2017. – P. 15. – arXiv:1706.03762

5. Большой обзор больших языковых моделей. URL: https://habr.com/ru/companies/gaz-is/articles/884410/ (дата обращения: 20.03.2025).

6. Cameron R. Wolfe. Graph-Based Prompting and Reasoning with Language Models. URL: https://cameronrwolfe.substack.com/p/graph-based-prompting-and-reasoning?open=false#%C2%A7the-transformer-from-top-to-bottom (дата обращения: 17.04.2025).

7. Raffel C., Shazeer N., Roberts A. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer // Journal of Machine Learning Research. 2020.

8. Предобработка данных. URL: https://huggingface.co/learn/llm-course/ru/chapter3/2?fw=pt (дата обращения: 15.04.2025).

9. Cameron R. Understanding and Using Supervised Fine-Tuning (SFT) for Language Models. URL: https://cameronrwolfe.substack.com/p/understanding-and-using-supervised (дата обращения: 18.04.2025).

10. Hu E. J., Shen Y., Wallis P., Allen-Zhu Z., Li Y., Wang S., Wang L., Chen W. LoRA: Low-Rank Adaptation of Large Language Models. 2021. arXiv. https://doi.org/10.48550/arXiv.2106.09685.

11. Liu S., Liu Z., Huang X., Dong P., Cheng K.-T. LLM-FP4: 4-Bit Floating-Point Quantized Transformers. arXiv. https://doi.org/10.48550/arXiv.2310.16836. -2023

12. Brown T.B., Mann B. Language Models are Few-Shot Learners // Advances in Neural Information Processing Systems. 2020.

13. William B. Dolan, Chris Brockett. Automatically Constructing a Corpus of Sentential Paraphrases // Proceedings of the Third International Workshop on Paraphrasing (IWP2005). 2005.

Об авторах

ПУГАЧЕВ Андрей Сергеевич, аспирант, инженер лаб. робототехнических систем
Институт проблем морских технологий им. академика М.Д. Агеева Дальневосточного отделения Российской академии наук
Адрес: 690091, Владивосток, ул. Суханова, 5А
Научные интересы: программирование в робототехнике, АНПА, ТНПА, нейронные сети, трансформеры, большие языковые модели (programming in robotics, AUV, ROV, neural networks, transformers, large language models)
E-mail: pugachev@marine.febras.ru
Тел.: 8(423)2-215-545, доб. 512

БОРОВИК Алексей Игоревич, к. т. н., ведущий научный сотрудник, зав. лаб. робототехнических систем
Институт проблем морских технологий им. академика М.Д. Агеева Дальневосточного отделения Российской академии наук
Адрес: 690091, Владивосток, ул. Суханова, 5А
Научные интересы: системы управления роботами, робототехнические программные платформы, АНПА, ТНПА, операционные системы, нейросети (robotics control systems, robotics software frameworks, AUV, ROV, operating systems, neural networks)
E-mail: alexey@borovik.me. **Тел.**: 8(423)2-215-545, доб. 509
ORCID: 0000-0002-9696-2751

DOI: 10.37102/1992-4429_2025_53_03_06

COMPARATIVE ANALYSIS OF NEURAL NETWORKS-TRANSFORMERS IN SOLVING THE PROBLEM OF AUV MISSION GENERATION

A.S. Pugachev, A.I. Borovik

The article discusses the task of automating the generation of mission programs for autonomous underwater vehicles (AUV) using neural networks. The main focus is on choosing the optimal model for converting commands in natural language into code in a specialized underwater research programming language (URPL). LLaMA 3.1 8B-instruct, Gemma 2 9B IT, Mistral 7B-instruct, Qwen 7B-instruct, Phi-4 3.8B models are compared according to the criteria of generation accuracy, execution speed, resistance to errors in input data and compactness of the model when solving the given problem. The analysis results showed that the LLaMA 3.1 8B-instruct and Phi-4 3.8B models demonstrate the best performance. However, LLaMA 3.1 stands out for its higher processing speed (56.08 tokens per second) and the ability to correctly interpret commands, which makes it preferable for the tasks of generating AUV missions. To improve the accuracy and adaptability of the model, an approach to additional training of the neural network on an extended dataset created using more powerful language models is proposed.

Keywords: AUV, neural networks, mission, AUV mission programming language, URPL.

References

1. Ageev M.D., Kasatkin B.A., Kiselev L.V., Molokov Ju.G., Nikiforov V.V., Rylov N.I. Avtomaticheskie podvodnye apparaty. Leningrad: Sudostroenie, 1981. 224 p. [In Russ.]

2. Inzarcev A.V., Kiselev L.V., Kostenko V.V., Matvienko Ju.V., Pavin A.M., Shherbatjuk A.F. Underwater robotics: systems, technologies, application. Vladivostok: IMTP FEB RAS. Vladivostok, 2018. 368 p. [In Russ.]

3. Pugachev, A. S., Borovik A.I. URPL – universal language for AUV mission programming. Underwater investigations and robotics. 2024. No. 3(49). P. 26–37. DOI 10.37102/1992-4429_2024_49_03_03. EDN GVSAMY. [In Russ.]

4. Vaswani A., Shazeer N., Parmar N., Uszkoreit J., Jones L., Gomez A. N., Kaiser Ł., Polosukhin I. Attention is All you Need // Advances in Neural Information Processing Systems 30 / I. Guyon, U. v. Luxburg, S. Bengio, H. Wallach, R. Fergus, S.V.N. Vishwanathan, R. Garnett – 2017. – P. 15. – arXiv:1706.03762

5. Bol'shoj obzor bol'shih jazykovyh modelej. URL: https://habr.com/ru/companies/gaz-is/articles/884410/ (date of access: 20.03.2025). [In Russ.]

6. Cameron R. Wolfe. Graph-Based Prompting and Reasoning with Language Models. URL: https://cameronrwolfe.substack.com/p/graph-based-prompting-and-reasoning?open=false#%C2%A7the-transformer-from-top-to-bottom (date of access: 17.04.2025).

7. Raffel C., Shazeer N., Roberts A. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer // Journal of Machine Learning Research. 2020.

8. Predobrabotka dannyh. URL: https://huggingface.co/learn/llm-course/ru/chapter3/2?fw=pt (date of access: 15.04.2025). [In Russ.]

9. Cameron R. Understanding and Using Supervised Fine-Tuning (SFT) for Language Models. URL: https://cameronrwolfe.substack.com/p/understanding-and-using-supervised (date of access: 18.04.2025).

10. Hu E. J., Shen Y., Wallis P., Allen-Zhu Z., Li Y., Wang S., Wang L., Chen W. LoRA: Low-Rank Adaptation of Large Language Models. 2021. arXiv. https://doi.org/10.48550/arXiv.2106.09685.

11. Liu S., Liu Z., Huang X., Dong P., Cheng K.-T. LLM-FP4: 4-Bit Floating-Point Quantized Transformers. arXiv. https://doi.org/10.48550/arXiv.2310.16836. 2023

12. Brown T.B., Mann B. Language Models are Few-Shot Learners // Advances in Neural Information Processing Systems. 2020.

13. William B. Dolan, Chris Brockett. Automatically Constructing a Corpus of Sentential Paraphrases // Proceedings of the Third International Workshop on Paraphrasing (IWP2005). 2005.

Information about authors

PUGACHEV Andrey Sergeevich, Postgraduate student, engineer at the laboratory of robotic systems
Institute of Marine Technology Problems, Far Eastern Branch of Russian Academy of Science
Address: Russia, 690091, Vladivostok, Sukhanova str., 5A
E-mail: pugachev@marine.febras.ru
Phone: +7(423)2-215-545, ext. 512

BOROVIK Alexey Igorevich, Candidate of Technical Sciences, Leading Researcher, Head of the Laboratory of Robotic Systems
Institute of Marine Technology Problems, Far Eastern Branch of Russian Academy of Science
Address: Russia, 690091, Vladivostok, Sukhanova str., 5A
E-mail: alexey@borovik.me. **Phone**: +7(423)2-215-545, ext. 509
ORCID: 0000-0002-9696-2751