МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«КУБАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
Факультет компьютерных технологий и прикладной математики

УТВЕРЖДАЮ:
Проректор по учебной работе,
качеству образования – первый
проректор
_____Хагуров Т.А.
подпись
« 29 » августа 2025 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ (МОДУЛЯ)

Б1. В.04 Промпт инжиниринг в профессиональной деятельности

Направление подготовки 01.03.02 Прикладная математика и информатика

Профиль Современные методы машинного обучения и компьютерного зрения

Форма обучения очная

Квалификация бакалавр

Краснодар 2025

Рабочая программа дисциплины «Промпт инжиниринг в профессиональной деятельности» составлена в соответствии с федеральным государственным образовательным стандартом высшего образования (ФГОС ВО) по направлению подготовки 01.03.02 Прикладная математика и информатика

Программу составил(и):
А.В. Коваленко, заведующий КАДИИ, д-р.тех.н., доцент
И.О. Фамилия, должность, ученая степень, ученое звание — подпись

Рабочая программа дисциплины утверждена на заседании центра искусственного интеллекта
протокол № 01 «28» августа 2025 г.
Руководитель центра ИИ Коваленко А.В.
подпись

Утверждена на заседании учебно-методической комиссии факультета компьютерных технологий и прикладной математики
протокол № 01 «28» августа 2025 г.
Председатель УМК факультета Коваленко А.В.
подпись

Рецензенты:
Мостовой Евгений Викторович, генеральный директор ООО «Портал-Юг», e-mail: mostovoy@portal-yug.ru

Луценко Евгений Вениаминович, доктор экономических наук, кандидат технических наук, профессор кафедры компьютерных технологий и систем Федерального государственного бюджетное образовательное учреждение высшего образования «Кубанский государственный аграрный университет имени И.Т. Трубилина», e-mail: prof.lutsenko@gmail.com

1 Цели и задачи изучения дисциплины (модуля)

1.1 Цель освоения дисциплины

Формирование у студентов базовых знаний, навыков и компетенций в области промпт-инжиниринга — ключевого направления взаимодействия с современными генеративными ИИ-моделями (LLM — Large Language Models), необходимого для профессиональной деятельности аналитика данных, AI-инженера, MLOps-специалиста и менеджера ИИ-проектов.

1.2 Задачи дисциплины

1. Ознакомить студентов с архитектурой и принципами работы генеративных моделей (ChatGPT, DeepSeek, Perplexity, Grok, Llama, GigaChat, YandexGPT, Qwen, Claude, Gemini, Manus и др.).
2. Научить базовым техникам промпт-инжиниринга: шаблоны, цепочки рассуждений, system prompts.
3. Развить навыки интеграции промптов в пайплайны решений прикладных задач.
4. Подготовить к созданию собственных промптов и анализу их эффективности.
5. Показать профессиональные кейсы и сценарии применения промптов в ролях Data Analyst, MLOps, AI Project Manager.

1.3 Место дисциплины (модуля) в структуре образовательной программы

Дисциплина «Промпт инжиниринг в профессиональной деятельности» относится к «Части, формируемая участниками образовательных отношений» Блока 1 «Дисциплины (модули)» учебного плана.

1.4 Профессиональные роли в структуре образовательной программы

Роль 1: Data Engineer (Инженер по данным)
Задачи:
- Проектирование и построение ETL-процессов
- Создание и оптимизация хранилищ данных
- Обеспечение качества и доступности данных
- Настройка инфраструктуры для обработки больших данных
- Интеграция разрозненных источников данных
- Работа с данными в области природопользования, медицины, связи и телекоммуникаций

Роль 2: ML Engineer (Инженер МО)
Задачи:
- Реализация ML-моделей в продуктивных системах
- Оптимизация производительности и масштабирование моделей
- Разработка ML-пайплайнов и автоматизация процессов
- Мониторинг качества моделей в продуктиве
- Интеграция ML-решений с бизнес-приложениями

Роль 3: MLOps (Специалист по эксплуатации ИИ)
Задачи:
- Автоматизация процессов обучения и развертывания моделей
- Мониторинг производительности ML-систем
- Управление версиями моделей и данных
- Обеспечение CI/CD для ML-проектов
- Оптимизация вычислительных ресурсов

1.5 Перечень планируемых результатов обучения по дисциплине (модулю), соотнесенных с планируемыми результатами освоения образовательной программы

Изучение данной учебной дисциплины направлено на формирование у обучающихся следующих компетенций:

Код и наименование индикатора*	Результаты обучения по дисциплине
ML-1 П Способен применять знания об истории развития и трендах современного ИИ для формулирования корректных постановок задач и поиска перспективных способов решения проблем с помощью ИИ.	
ML-1.1 Позиционирует собственную задачу в заданной области знания с точки зрения трендов современного искусственного интеллекта	Анализирует и сопоставляет задачу с современными трендами, выделяет специфику задачи в контексте последних достижений ИИ
ML-2 П Способен применять фундаментальные принципы и методы машинного обучения включая подготовку данных оценку качества моделей и работу с признаками	
ML-2.1 Различает основные типы задач машинного обучения и применяет на практике принципы их решения	Выбирает и обосновывает методы решения задач машинного обучения с учётом характеристик данных и бизнес-контекста, настраивает базовые модели и проводит их оценку
ML-2.2 Применяет методы предварительной обработки данных и работы с признаками	Владеет методами feature engineering: отбор создание и преобразование признаков.
DL-2 Б Способен применять и (или) разрабатывать современные архитектуры генеративных глубоких сетей	
DL-2.1 Применяет известные архитектуры генеративных глубоких нейронных сетей для решения прикладной задачи (генерация текста, генерация изображений по тексту, синтез речи и т.д.), при необходимости проводя дообучение на наборах данных	Умеет использовать популярные генеративные модели (GPT, Stable Diffusion, VQ-VAE) через API или готовые реализации. Запускает инференс на стандартных задачах (генерация текста по промпту, создание изображений). Работает с базовыми параметрами генерации (temperature, top-k sampling). Подготавливает данные для дообучения (токенизация текста, нормализация изображений). Форматирует данные под требования модели (например, промпты для тексто-изображение моделей).
LLM-5 П Организует взаимодействие с генеративными моделями через проектирование, анализ и применение промптов	
LLM-5.1 Использует базовые шаблоны промптов	Выбирает и адаптирует шаблоны под задачу
LLM-5.2 Встраивает промпты в пайплайн взаимодействия	Применяет цепочки (Chain of Thought) и условную логику
LLM-5.4 Разрабатывает дизайн и структуру промптов	Оптимизирует промпты под точность, длину, уменьшение галлюцинаций

2. Структура и содержание дисциплины

2.1 Распределение трудоёмкости дисциплины по видам работ

Общая трудоёмкость дисциплины составляет 2 зач. ед. (72 часов), их распределение по видам работ представлено в таблице

Вид учебной работы	Всего часов	Семестры (часы)					
		1					

Контактная работа, в том числе:		**57,2**	**57,2**					
Аудиторные занятия (всего):		**50**	**50**					
Занятия лекционного типа		**16**	16					
Лабораторные занятия		**34**	34					
Занятия семинарского типа (семинары, практические занятия)								
Иная контактная работа:		**7,2**	**7,2**					
Контроль самостоятельной работы (КСР)		**2**	2					
Промежуточная аттестация (ИКР)		**5,2**	5,2					
Самостоятельная работа, в том числе:		**14,8**	**14,8**					
Подготовка курсовой работы		**5**	5					
Проработка учебного (теоретического) материала		**5**	5					
Выполнение индивидуальных заданий (подготовка сообщений, презентаций)		**4,8**	4,8					
Реферат								
Подготовка к текущему контролю								
Контроль:								
Подготовка к экзамену								
Общая трудоемкость	**час.**	**72**	**72**					
	в том числе контактная работа	**57,2**	**57,2**					
	зач. ед	**2**	**2**					

2.2 Структура дисциплины

Распределение видов учебной работы и их трудоемкости по разделам дисциплины.

Разделы (темы) дисциплины, изучаемые в 1 семестре

№	Наименование разделов (тем)	Количество часов				
		Всего	Аудиторная работа			Внеаудиторная работа
			Л	ПЗ	ЛР	СРС
1	**2**	**3**	**4**	**5**	**6**	**7**
1.	Введение в генеративный ИИ и LLM	4	2		1	1
2.	Архитектуры генеративных моделей	5	2		2	1
3.	Принципы промпт-инжиниринга	7	2		4	1
4.	Шаблоны и стратегии промптов	7	2		4	1
5.	Создание пайплайнов взаимодействия с ИИ	8	2		4	2
6.	Интеграция промптов в бизнес-кейсы	6			4	2
7.	Отладка, тестирование и метрики качества	8	2		4	2
8.	Мультимодальные промпты и интерфейсы	8	2		4	2
9.	Этические аспекты и защита от генеративных рисков	6,8	2		2	2,8
ИТОГО по разделам дисциплины		**64,8**	**16**		**34**	**14,8**
Контроль самостоятельной работы (КСР)		2				
Промежуточная аттестация (ИКР)		5,2				
Подготовка к текущему контролю						
Общая трудоемкость по дисциплине		**72**				

Примечание: Л – лекции, ПЗ – практические занятия/семинары, ЛР – лабораторные занятия, СРС – самостоятельная работа студента

2.3 Содержание разделов (тем) дисциплины

2.3.1 Занятия лекционного типа

№	Наименование раздела (темы)	Содержание раздела (темы)	Форма текущего контроля
1	2	3	4
1.	Введение в генеративный ИИ и LLM	История развития и основные тренды современного ИИ. Понятие генеративного ИИ. Различие между дискриминативными и генеративными моделями. История и развитие LLM: от GPT-2 до GPT-5, DeepSeek, Perplexity, Grok, Llama, GigaChat, YandexGPT, Qwen, Claude, Gemini, Manus. Роль генеративных моделей в аналитике, разработке и управлении проектами. Обзор ключевых применений (ассистенты, генерация текста, анализ данных, автоматизация).	ЛР
2.	Архитектуры генеративных моделей	Transformer, Attention, Positional Encoding — базовые механизмы. Сравнение ChatGPT, DeepSeek, Perplexity, Grok, Llama, GigaChat, YandexGPT, Qwen, Claude, Gemini, Manus, Diffusion-моделей. Архитектурные особенности LLM для текстов и моделей для изображений/звука. Обзор инструментов: Hugging Face, OpenAI API, Diffusers.	ЛР
3.	Принципы промпт-инжиниринга	Что такое промпт и зачем он нужен. Промпт как программный интерфейс к модели. Контекст, токенизация, температурные параметры и параметры вывода. Основные ошибки в промптах и их влияние на генерацию.	ЛР
4.	Шаблоны и стратегии промптов	Zero-shot, One-shot, Few-shot, Chain-of-thought prompting. Шаблоны: Q&A, таблицы, списки, role-based prompting. Использование system prompts и инструкций. Интерактивные цепочки промптов.	ЛР
5.	Создание пайплайнов взаимодействия с ИИ	Сценарии автоматизации: от простых скриптов до web-приложений. Интеграция промптов в чат-ботов, дашборды и REST API. Использование Python (FastAPI), Node.js, Postman, cURL для тестирования. Пример пайплайна: ввод → промпт → вызов LLM → результат → логика обработки.	ЛР
7.	Отладка, тестирование и метрики качества	Методы оценки результатов генерации: BLEU, ROUGE, METEOR, Self-BLEU. A/B тестирование промптов. Анализ ошибок модели, диагностика неожиданных ответов. Версионирование промптов и контроль изменений.	ЛР
8.	Мультимодальные промпты и интерфейсы	Работа с изображениями, аудио, видео: от CLIP и DALL·E до Gemini. Комбинированные промпты: "нарисуй + опиши", "расскажи о фото".	ЛР

№	Наименование раздела (темы)	Содержание раздела (темы)	Форма текущего контроля
1	2	3	4
		Примеры использования: генерация инфографики, дизайн, synthetic data. Инструменты: Stable Diffusion, Midjourney, Whisper, Gemini.	
9.	Этические аспекты и защита от генеративных рисков	Проблемы генерации токсичного, фейкового или дискриминирующего контента. Методы фильтрации и безопасного взаимодействия. Этические гайды от OpenAI, DeepMind, UNESCO. Понятие "responsible AI" и профессиональная этика промпт-инженера.	ЛР

Примечание: ЛР – отчет/защита лабораторной работы, КП - выполнение курсового проекта, КР - курсовой работы, РГЗ - расчетно-графического задания, Р - написание реферата, Э - эссе, К - коллоквиум, Т – тестирование, РЗ – решение задач.

2.3.2 Занятия семинарского типа

Не предусмотрены

Примечание: ЛР – отчет/защита лабораторной работы, КП - выполнение курсового проекта, КР - курсовой работы, РГЗ - расчетно-графического задания, Р - написание реферата, Э - эссе, К - коллоквиум, Т – тестирование, РЗ – решение задач.

2.3.3 Лабораторные занятия

№	Наименование раздела (темы)	Наименование лабораторных работ	Форма текущего контроля
1	2	3	4
1.	Введение в генеративный ИИ и LLM	Регистрация и запуск ChatGPT, DeepSeek, Perplexity, Grok, Llama, GigaChat, YandexGPT, Qwen, Claude, Gemini, Manus, ввод различных вопросов: факты, мнения, креативные задачи. Сравнение ответов при одинаковом запросе, повторяемость. Анализ структуры ответа. Пример: задать один и тот же вопрос в стиле "студента", "профессора", "менеджера".	ЛР
2.	Архитектуры генеративных моделей	Сравнение архитектур ChatGPT, DeepSeek, Perplexity, Grok, Llama, GigaChat, YandexGPT, Qwen, Claude, Gemini, Manus. Подключение к Hugging Face и запуск модели через Spaces. Анализ числа параметров, скорости, качества генерации. Исследование конфигурации модели. Использование интерфейса Inference API.	ЛР
3.	Принципы промпт-инжиниринга	Цель: Освоить работу с параметрами генерации и структурой промптов. Содержание: Температура, Top-p, Max	ЛР

№	Наименование раздела (темы)	Наименование лабораторных работ	Форма текущего контроля
1	2	3	4
		tokens, Presence/penalty. Проведение серии запросов с изменением параметров и анализом результатов. Формулировка промптов в виде инструкций, вопросов, ролевых моделей.	
4.	Шаблоны и стратегии промптов	Сравнение эффективности Zero-shot vs Few-shot. Построение цепочки размышлений в задачах логики (напр. задачки на монеты, путешествия). Создание промптов с примерами.	ЛР
5.	Создание пайплайнов взаимодействия с ИИ	Установка FastAPI, создание эндпоинта POST /generate. Обработка входного текста, вызов модели (через OpenAI API или Hugging Face). Форматирование вывода. Тестирование с Postman или curl.	ТЛР
6.	Интеграция промптов в бизнес-кейсы	Цель: Демонстрировать прикладную значимость промпт-инжиниринга. Сценарии: генерация email-ответов, кратких новостей, отчётов по данным. Формализация задачи → построение промпта → оценка результата.	ЛР
7.	Отладка, тестирование и метрики качества	Проведение серии генераций, подсчёт BLEU/ROUGE (с использованием готовых библиотек). Анализ точности и вариативности. Проведение A/B тестирования: два промпта — один набор входных данных. Создание таблицы с метриками и субъективной оценкой.	ЛР
8.	Мультимодальные промпты и интерфейсы	Работа с Stable Diffusion (через web-интерфейс или Colab). Создание описания → генерация изображения. Работа с обратной задачей: CLIP/BLIP — генерация подписи к изображению.	ЛР
9.	Этические аспекты и защита от генеративных рисков	Генерация спорных текстов (фактчекинг, токсичность, дискриминация). Примеры использования фильтров (moderation API). Создание промпта для этичного взаимодействия. Анализ потенциальных нарушений.	ЛР
10	Курсовая работа и защита	Цель: Продемонстрировать понимание проектирования промптов и их применения в профессиональной деятельности	ЛР

Примечание: ЛР – отчет/защита лабораторной работы, КП - выполнение курсового проекта, КР - курсовой работы, РГЗ - расчетно-графического задания, Р - написание реферата, Э - эссе, К - коллоквиум, Т – тестирование, РЗ – решение задач.

2.3.4 Примерная тематика курсовых работ (проектов)

1. Разработка промптов для генерации Python/C-кода на основе текстового описания задачи

2. Сравнение качества генерации программ с помощью различных промптов (zero-shot vs few-shot)

3. Генерация кода циклов и условий: промпты для типовых задач (факториал, Фибоначчи и т.д.)

4. Создание промптов для автокомментирования кода на Python/C

5. Разработка ассистента на базе LLM для решения задач по основам программирования

6. Анализ промптов для генерации функций сортировки и обработки массивов

7. Построение цепочек промптов для отладки кода и поиска ошибок

8. Автоматическая генерация тестов к простым Python/C -функциям с помощью LLM

9. Промпты для генерации кода с различной степенью детализации и пояснений

10. Сравнение выходов от ChatGPT и GitHub Copilot на одинаковые промпты по Python/C

11. Разработка промптов для вычисления пределов и производных с пояснением шагов

12. Промпты для генерации решений задач по интегралам (неопределённые и определённые)

13. Промпты, обучающие LLM решать уравнения и неравенства (включая пошаговые рассуждения)

14. Сравнение точности и корректности вывода решений по математическому анализу

15. Разработка промптов для визуализации графиков функций

16. Анализ ошибок при генерации решений по математическому анализу: примеры и способы повышения точности

17. Использование промптов для генерации кратких справок и теоретических сводок по темам

18. Построение промптов, объясняющих суть пределов, производных и их применений

19. Создание промптов-репетиторов: моделирование учебного диалога по математическому анализу

20. Интеграция LLM в Jupyter-ноутбуки для помощи в решении задач по математическому анализу

21. Генерация промптов для нахождения скалярного и векторного произведения

22. Построение промптов для расчета длины, нормализации и угла между векторами

23. Разработка промптов для решений задач с матрицами: транспонирование, умножение

24. Промпты для объяснения линейной зависимости и базиса

25. Моделирование диалога между студентом и LLM по решению СЛАУ

26. Использование промптов для визуализации векторов и линейных преобразований

27. Создание обучающих карточек по темам векторной алгебры с помощью LLM

28. Разработка тестов по темам линейной алгебры через генерацию LLM

29. Промпты для составления пошаговых решений по методу Гаусса

30. Сравнение промптов для символьного и численного решения задач

31. Создание учебного бота для 1 курса: вопросы по Python /C, математическому анализу и векторной алгебре

32. Исследование эффективности генерации задач с решениями для подготовки к зачету

33. Сценарии применения промптов в электронных учебниках и курсах

34. Генерация задач олимпиадного уровня и пошаговых решений к ним

35. Автоматическое составление шпаргалки по темам курса

36. Промпты для генерации викторин и мини-игр по математике и программированию

37. Создание LLM-ассистента, который «объясняет, как преподаватель» — стилизация ответа

38. Разработка промптов для генерации простых визуальных задач по аналитической геометрии

39. Исследование возможности генерации контрпримеров и логических ловушек в задачах

40. Разработка генератора практикумов для подготовки к экзамену на базе промптов

41. Разработка промптов для автоматического анализа пользовательских отзывов

42. Построение промптов для генерации запросов на естественном языке

43. Сравнительный анализ качества аналитических отчетов, созданных с помощью LLM

44. Использование промптов для обобщения больших текстовых массивов в аналитике

45. Автоматическая визуализация данных по текстовому описанию с использованием LLM

46. Промпт-интерфейсы для объяснения метрик и KPI

47. Создание ассистента для анализа отчетности и подсветки ключевых трендов

48. Разработка шаблонов промптов для интерпретации результатов A/B-тестов

49. Генерация рекомендаций по улучшению метрик продукта на основе текстового отчета

50. Сравнение подходов chain-of-thought и few-shot prompting в аналитике данных

51. Встраивание LLM в пайплайн мониторинга моделей машинного обучения

52. Промпты для генерации документации ML-проекта по коду и логам

53. Промпты для генерации тестов и проверок качества моделей

54. Автоматическая генерация DAG-пайплайна для ML-задачи по описанию

55. Разработка системы версионирования и контроля промптов в продуктивной среде

56. Использование LLM для анализа логов и диагностики инцидентов

57. Создание инструментов отладки и наблюдаемости через генеративные интерфейсы

58. Генерация REST API документации по описанию модели

59. Построение ассистента для настройки гиперпараметров моделей

60. Интеграция LLM в CI/CD пайплайн проекта машинного обучения

61. Промпт-интерфейс для составления технического задания на ИИ-систему

62. Автоматизация составления project charter с помощью LLM

63. Промпт-интерфейсы для анализа рисков ИИ-проекта

64. Разработка шаблонов генерации ROI-оценок и бизнес-кейсов

65. Внедрение LLM-ассистента в трекинг задач и задач планирования спринтов

66. Использование LLM для анализа обратной связи по результатам ИИ-внедрения

67. Построение system prompt'ов для сценариев взаимодействия с заказчиком

68. Генерация текстов презентаций и отчетов по проекту ИИ

69. Исследование эффективности LLM в роли фасилитатора проектных совещаний

70. Создание мета-промптов для быстрой адаптации под задачи клиента

71. Промпты для генерации UX-интерфейсов по текстовому описанию

72. Разработка промптов для генерации изображений в корпоративной стилистике

73. Мультимодальные промпты: генерация инфографики по аналитическому описанию

74. Промпт-интерфейсы для создания сценариев обучения персонала

75. Генерация юридических документов на основе текстовых описаний требований

76. Разработка промптов для создания диалогов и FAQ-систем

77. Промпт-интерфейс для автоматической генерации вакансий и резюме

78. Промпты для генерации стратегий продвижения продуктов на основе описания ниши

79. Исследование надёжности и повторяемости вывода при генерации технических текстов

80. Этические фильтры и их влияние на точность генерации: отбор и тестирование промптов

2.4 Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине (модулю)

№	Вид СРС	Перечень учебно-методического обеспечения дисциплины по выполнению самостоятельной работы
1	2	3
1	Изучение теоретического материала	Методические указания по организации самостоятельной работы студентов, утвержденные УМК ФКТиПМ, протокол №1 от 29.08.2025
2	Решение задач	Методические указания по организации самостоятельной работы студентов, утвержденные УМК ФКТиПМ, протокол №1 от 29.08.2025

Учебно-методические материалы для самостоятельной работы обучающихся из числа инвалидов и лиц с ограниченными возможностями здоровья (ОВЗ) предоставляются в формах, адаптированных к ограничениям их здоровья и восприятия информации:

Для лиц с нарушениями зрения:

– в печатной форме увеличенным шрифтом,

– в форме электронного документа,

– в форме аудиофайла,

– в печатной форме на языке Брайля.

Для лиц с нарушениями слуха:

– в печатной форме,

– в форме электронного документа.

Для лиц с нарушениями опорно-двигательного аппарата:

– в печатной форме,

– в форме электронного документа,

– в форме аудиофайла.

Данный перечень может быть конкретизирован в зависимости от контингента обучающихся.

3. Образовательные технологии

В соответствии с требованиями ФГОС в программа дисциплины предусматривает использование в учебном процессе следующих образовательные технологии: чтение лекций с использованием мультимедийных технологий; метод малых групп, разбор практических задач и кейсов.

При обучении используются следующие образовательные технологии:

– Технология коммуникативного обучения – направлена на формирование коммуникативной компетентности студентов, которая является базовой, необходимой для адаптации к современным условиям межкультурной коммуникации.

– Технология разноуровневого (дифференцированного) обучения – предполагает осуществление познавательной деятельности студентов с учётом их индивидуальных

способностей, возможностей и интересов, поощряя их реализовывать свой творческий потенциал. Создание и использование диагностических тестов является неотъемлемой частью данной технологии.

– Технология модульного обучения – предусматривает деление содержания дисциплины на достаточно автономные разделы (модули), интегрированные в общий курс.

– Информационно-коммуникационные технологии (ИКТ) - расширяют рамки образовательного процесса, повышая его практическую направленность, способствуют интенсификации самостоятельной работы учащихся и повышению познавательной активности. В рамках ИКТ выделяются 2 вида технологий:

– Технология использования компьютерных программ – позволяет эффективно дополнить процесс обучения языку на всех уровнях.

– Интернет-технологии – предоставляют широкие возможности для поиска информации, разработки научных проектов, ведения научных исследований.

– Технология индивидуализации обучения – помогает реализовывать личностно-ориентированный подход, учитывая индивидуальные особенности и потребности учащихся.

– Проектная технология – ориентирована на моделирование социального взаимодействия учащихся с целью решения задачи, которая определяется в рамках профессиональной подготовки, выделяя ту или иную предметную область.

– Технология обучения в сотрудничестве – реализует идею взаимного обучения, осуществляя как индивидуальную, так и коллективную ответственность за решение учебных задач.

– Игровая технология – позволяет развивать навыки рассмотрения ряда возможных способов решения проблем, активизируя мышление студентов и раскрывая личностный потенциал каждого учащегося.

– Технология развития критического мышления – способствует формированию разносторонней личности, способной критически относиться к информации, умению отбирать информацию для решения поставленной задачи.

– Комплексное использование в учебном процессе всех вышеназванных технологий стимулируют личностную, интеллектуальную активность, развивают познавательные процессы, способствуют формированию компетенций, которыми должен обладать будущий специалист.

Основные виды интерактивных образовательных технологий включают в себя:

– работа в малых группах (команде) - совместная деятельность студентов в группе под руководством лидера, направленная на решение общей задачи путём творческого сложения результатов индивидуальной работы членов команды с делением полномочий и ответственности;

– проектная технология - индивидуальная или коллективная деятельность по отбору, распределению и систематизации материала по определенной теме, в результате которой составляется проект;

– анализ конкретных ситуаций - анализ реальных проблемных ситуаций, имевших место в соответствующей области профессиональной деятельности, и поиск вариантов лучших решений;

– развитие критического мышления – образовательная деятельность, направленная на развитие у студентов разумного, рефлексивного мышления, способного выдвинуть новые идеи и увидеть новые возможности.

Подход разбора конкретных задач и ситуаций широко используется как преподавателем, так и студентами во время лекций, лабораторных занятий и анализа результатов самостоятельной работы. Это обусловлено тем, что при исследовании и решении каждой конкретной задачи имеется, как правило, несколько методов, а это требует разбора и оценки целой совокупности конкретных ситуаций.

Семестр	Вид занятия	Используемые интерактивные образовательные технологии	количество интерактивных часов
1	ЛР	Практические занятия в режимах взаимодействия «преподаватель – студент» и «студент – студент»	12
Итого			**12**

Примечание: Л – лекции, ПЗ – практические занятия/семинары, ЛР – лабораторные занятия, СРС – самостоятельная работа студента

Темы, задания и вопросы для самостоятельной работы призваны сформировать навыки поиска информации, умения самостоятельно расширять и углублять знания, полученные в ходе лекционных и практических занятий.

Подход разбора конкретных ситуаций широко используется как преподавателем, так и студентами при проведении анализа результатов самостоятельной работы.

Для лиц с ограниченными возможностями здоровья предусмотрена организация консультаций с использованием электронной почты.

Для лиц с нарушениями зрения:

– в печатной форме увеличенным шрифтом,

– в форме электронного документа.

Для лиц с нарушениями слуха:

– в печатной форме,

– в форме электронного документа.

Для лиц с нарушениями опорно-двигательного аппарата:

– в печатной форме,

– в форме электронного документа.

Для лиц с ограниченными возможностями здоровья предусмотрена организация консультаций с использованием электронной почты.

Данный перечень может быть конкретизирован в зависимости от контингента обучающихся.

4. Оценочные и методические материалы

4.1 Оценочные средства для текущего контроля успеваемости и промежуточной аттестации

Оценочные средства предназначены для контроля и оценки образовательных достижений обучающихся, освоивших программу учебной дисциплины «Промпт инжиниринг в профессиональной деятельности».

Оценочные средства включает контрольные материалы для проведения **текущего контроля** в форме тестовых заданий, кейсов и **промежуточной аттестации** в форме вопросов и заданий к **зачету**.

Оценочные средства для инвалидов и лиц с ограниченными возможностями здоровья выбираются с учетом их индивидуальных психофизических особенностей.

– при необходимости инвалидам и лицам с ограниченными возможностями здоровья предоставляется дополнительное время для подготовки ответа на зачете;

– при проведении процедуры оценивания результатов обучения инвалидов и лиц с ограниченными возможностями здоровья предусматривается использование технических средств, необходимых им в связи с их индивидуальными особенностями;

– при необходимости для обучающихся с ограниченными возможностями здоровья и инвалидов процедура оценивания результатов обучения по дисциплине может проводиться в несколько этапов.

Процедура оценивания результатов обучения инвалидов и лиц с ограниченными возможностями здоровья по дисциплине (модулю) предусматривает предоставление информации в формах, адаптированных к ограничениям их здоровья и восприятия информации:

Для лиц с нарушениями зрения:

– в печатной форме увеличенным шрифтом,

– в форме электронного документа.

Для лиц с нарушениями слуха:

– в печатной форме,

– в форме электронного документа.

Для лиц с нарушениями опорно-двигательного аппарата:

– в печатной форме,

– в форме электронного документа.

Данный перечень может быть конкретизирован в зависимости от контингента обучающихся.

Структура оценочных средств для текущей и промежуточной аттестации

№ п/п	Контролируемые разделы (темы) дисциплины	Код контролируемой компетенции (или ее части)	Наименование оценочного средства	
			Текущий контроль	Промежуточная аттестация
1	Введение в генеративный ИИ и LLM	ML-1.1, ML-2.1	*Лабораторная работа №1*	*Вопросы к зачету 1-4*
2	Архитектуры генеративных моделей	ML-1.1, DL-2.1	*Лабораторная работа №2*	*Вопросы к зачету 5-8*
3	Принципы промпт-инжиниринга	ML-2.1, LLM-5.1	*Лабораторная работа №3*	*Вопросы к зачету 9-12*
4	Шаблоны и стратегии промптов	LLM-5.1, LLM-5.4	*Лабораторная работа №4*	*Вопросы к зачету 13-16*
5	Создание пайплайнов взаимодействия с ИИ	LLM-5.2,	*Лабораторная работа №5*	*Вопросы к зачету 17-20*
6	Интеграция промптов в бизнес-кейсы	LLM-5.4	*Лабораторная работа №6*	*Вопросы к зачету 21-24* *Проект с кейсом от индустриального партнера*
7	Отладка, тестирование и метрики качества	ML-2.2 LLM-3.6	*Лабораторная работа №7*	*Вопросы к зачету 25-28*
8	Мультимодальные промпты и интерфейсы	DL-2.1, LLM-5.4	*Лабораторная работа №8*	*Вопросы к зачету 29-32* *Проект с кейсом от индустриального партнера*
9	Этические аспекты и защита от генеративных рисков	ML-1.1	*Лабораторная работа №9*	*Вопросы к зачету 33-36*
10	Курсовая работа и защита	LLM-5.1, LLM-5.2, LLM-5.4, ML-2.1–2.2, DL-2.1	*Курсовая работа и презентация*	*Вопросы к зачету 37-40* *Проект с кейсом от индустриального партнера*

Показатели, критерии и шкала оценки сформированных компетенций

Соответствие **<u>пороговому уровню</u>** освоения компетенций планируемым результатам обучения и критериям их оценивания (оценка: **зачтено**):

ML-1 П ***Способен применять знания об истории развития и трендах современного ИИ для формулирования корректных постановок задач и поиска перспективных способов решения проблем с помощью ИИ.***

Знает историю развития и основные тренды современного ИИ.
Позиционирует собственную задачу в заданной области знания с точки зрения трендов современного искусственного интеллекта.
Анализирует и сопоставляет задачу с современными трендами, выделяет специфику задачи в контексте последних достижений ИИ.
Оценивает конкурирующие решения и разработки с точки зрения трендов современного искусственного интеллекта
Оценивает конкурентные решения с учётом современных трендов (например, использование современных архитектур, подходов к интерпретируемости, устойчивости, энергоэффективности), анализирует преимущества и ограничения.

ML-2 П ***Способен применять фундаментальные принципы и методы машинного обучения включая подготовку данных оценку качества моделей и работу с признаками***

Различает основные типы задач машинного обучения и применяет на практике принципы их решения.
Выбирает и обосновывает методы решения задач машинного обучения с учётом характеристик данных и бизнес-контекста, настраивает базовые модели и проводит их оценку.
Применяет методы предварительной обработки данных и работы с признаками Владеет методами feature engineering: отбор создание и преобразование признаков.
Решает проблемы несбалансированных данных и оценивает качество моделей.
Применяет различные типы кросс-валидации Оценивает качество моделей с учетом bias-variance trade-off.

DL-2 Б ***Способен применять и (или) разрабатывать современные архитектуры генеративных глубоких сетей***

Применяет известные архитектуры генеративных глубоких нейронных сетей для решения прикладной задачи (генерация текста, генерация изображений по тексту, синтез речи и т.д.), при необходимости проводя дообучение на наборах данных.
Умеет запускать инференс на готовых генеративных моделях (Stable Diffusion, GPT). Проводит базовый fine-tuning предобученных моделей по готовым рецептам.
Использует стандартные пайплайны обучения из библиотек (Hugging Face, Diffusers). Подготавливает данные для дообучения (нормализация изображений, токенизация текста).
Применяет базовые техники аугментации данных. Упаковывает модели в Docker-контейнеры. Создает простые REST API для инференса (Flask/FastAPI).
Имплементирует известные архитектуры генеративных сетей, реализует пайплайны их обучения на датасетах и вывод генеративных моделей в продуктивную среду.

	Адаптирует существующие генеративные архитектуры (GAN, VAE, Diffusion) под специфические задачи. Проводит ablation studies компонентов моделей. Реализует идеи из современных статей (на уровне воспроизведения). Предлагает улучшения базовых подходов (новые функции потерь, регуляризации). Умеет экспериментировать с техниками стабилизации обучения.
LLM-5 П	***Организует взаимодействие с генеративными моделями через проектирование, анализ и применение промптов*** Использует базовые шаблоны промптов. Применяет цепочки (Chain of Thought) и условную логику. Встраивает промпты в пайплайн взаимодействия. Управляет параметрами генерации для контроля результата Разрабатывает дизайн и структуру промптов. Настраивает system prompts и вводит ограничения. Анализирует и отлаживает промпты. Интегрирует промпты для мультимодальной генерации. Применяет мультимодальные промпты. Организует хранение и версионирование

Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций в процессе освоения образовательной программы

Практические кейсы по тематике лабораторных работ

Лабораторная работа №1: Введение в генеративный ИИ и LLM

Кейс: «Ассистент-справочник для студента 1 курса»

Задача: Сформулируйте 5 различных промптов, которые могли бы использовать первокурсники для генерации справок по основам программирования, математике и векторной алгебре.

Что нужно сделать:

Провести генерацию через ChatGPT, DeepSeek, Perplexity, Grok, Llama, GigaChat, YandexGPT, Qwen, Claude, Gemini, Manus;

Оценить релевантность и полноту ответов;

Сделать выводы о влиянии формулировки промпта.

Инструменты: ChatGPT, DeepSeek, Perplexity, Grok, Llama, GigaChat, YandexGPT, Qwen, Claude, Gemini, Manus (markdown-отчёт), MS Word или Jupyter

Лабораторная работа №2: Архитектуры генеративных моделей

Кейс: «Выбор модели для создания образовательного ассистента»

Задача: Сравнить архитектуры GPT-2, GPT-3.5, GPT-4, DeepSeek, Perplexity, Grok и BERT с точки зрения применимости для генерации учебных материалов.

Что нужно сделать:

Изучить документацию Hugging Face;

Подключить модели через интерфейс или API;

Провести генерацию по одному и тому же сценарию;

Сравнить результаты по качеству и скорости.

Инструменты: Hugging Face, OpenAI Playground, Jupyter

Лабораторная работа №3: Принципы промпт-инжиниринга

Кейс: «Ответ в стиле преподавателя»

Задача: Разработать серию промптов, нацеленных на генерацию ответов в разных стилях (официальный, дружелюбный, краткий).

Что нужно сделать:

Сравнить результат генерации при использовании разных system prompt'ов;

Выделить ключевые фразы, влияющие на стиль;

Подготовить сравнительный анализ.

Инструменты: OpenAI Playground, DeepSeek, ChatGPT, Perplexity, Grok таблица Excel или Jupyter

Лабораторная работа №4: Шаблоны и стратегии промптов

Кейс: «Генерация ответа на вопрос по математическому анализу»

Задача: Решить задачу на определённый интеграл через zero-shot, few-shot и chain-of-thought prompting.

Что нужно сделать:

Написать по 3 промпта для одной задачи;

Провести генерацию;

Сравнить ход рассуждений модели.

Инструменты: ChatGPT, DeepSeek, Perplexity, Grok, Llama, GigaChat, YandexGPT, Qwen, Claude, Gemini, Manus, Google Docs (отчёт), Jupyter

Лабораторная работа №5: Создание пайплайнов взаимодействия с ИИ

Кейс: «Мини-сервис по генерации текстов»

Задача: Реализовать веб-интерфейс, который отправляет текстовый запрос к модели и выводит результат.

Что нужно сделать:

Настроить FastAPI эндпоинт;

Организовать отправку запроса к OpenAI API;

Протестировать ответы модели через curl/Postman.

Инструменты: Python, FastAPI, OpenAI API, curl/Postman

Лабораторная работа №6: Интеграция промптов в бизнес-кейсы

Кейс: «Генератор email для отдела продаж»

Задача: Разработать серию промптов, автоматизирующих подготовку писем клиентам.

Что нужно сделать:

Выделить типовые шаблоны писем;

Составить промпты под разные ситуации (приветствие, напоминание, предложение);

Провести тестирование на релевантность.

Инструменты: ChatGPT, DeepSeek, Perplexity, Grok, Llama, GigaChat, YandexGPT, Qwen, Claude, Gemini, Manus, таблица Excel, Markdown-документ

Лабораторная работа №7: Отладка, тестирование и метрики качества

Кейс: «Промпты для подготовки к зачету»

Задача: Оценить эффективность двух разных промптов, создающих объяснение темы по векторной алгебре.

Что нужно сделать:

Сформировать два промпта;

Оценить результат с точки зрения полноты, точности, ясности;

Рассчитать BLEU или ROUGE (с помощью библиотеки).

Инструменты: ChatGPT, DeepSeek, Perplexity, Grok, Llama, GigaChat, YandexGPT, Qwen, Claude, Gemini, Manus, nltk/rouge-score, Jupyter

Лабораторная работа №8: Мультимодальные промпты и интерфейсы

Кейс: «Обложка для онлайн-курса»

Задача: Сгенерировать изображение и сопроводительный текст для онлайн-курса по программированию.

Что нужно сделать:

Написать промпт для Stable Diffusion (через web-интерфейс);

Сформулировать title и описание для страницы курса;

Проверить соответствие изображения и текста.

Инструменты: Stable Diffusion WebUI / Leonardo.ai / DALL·E, DeepSeek, ChatGPT, Perplexity, Grok

Лабораторная работа №9: Этические аспекты и защита от генеративных рисков

Кейс: «Безопасный учебный чат-бот»

Задача: Проанализировать поведение модели при провокационных запросах. Разработать корректные формулировки, минимизирующие риски.

Что нужно сделать:

Сформулировать 5 спорных/пограничных запросов;

Оценить поведение модели;

Предложить промпты для безопасной генерации.

Инструменты: ChatGPT, DeepSeek, Perplexity, Grok, Llama, GigaChat, YandexGPT, Qwen, Claude, Gemini, Manus, OpenAI Moderation API (по желанию), Markdown-отчёт

Лабораторная работа №10: Курсовая работа

Кейс: «Проект промпт-интерфейса под задачу студента»

Задача: Самостоятельно выбрать задачу (по Python/C, математике, бизнесу) и создать ассистента на базе LLM.

Что нужно сделать:

Сформировать промпты и тест-кейсы;

Реализовать интерфейс (CLI/Web/Telegram-бот);

Подготовить документацию и провести защиту.

Инструменты: OpenAI API, Python, FastAPI/Gradio/Streamlit, Markdown/Word

Пример лабораторной работы

Лабораторная работа №3: Принципы промпт-инжиниринга

Название: Влияние параметров генерации и структуры промпта на результат

Цель:

Научиться проектировать эффективные промпты и понимать влияние параметров генерации на выход LLM.

Задачи:

1. Сравнить генерацию текста при использовании разных структур промпта (вопрос, инструкция, ролевая модель).
2. Изменить параметры генерации (temperature, top_p, max_tokens) и проанализировать результат.
3. Сделать выводы о принципах точного управления генерацией.

Ожидаемые результаты:

- Умение формировать промпты разных типов.
- Понимание ключевых параметров генерации.
- Навык анализа влияния параметров на результат.

Инструменты и библиотеки:

- ChatGPT, DeepSeek, Perplexity, Grok, Llama, GigaChat, YandexGPT, Qwen, Claude, Gemini, Manus / OpenAI Playground
- Python (опционально, через OpenAI API)
- Jupyter Notebook / Google Colab

Исходные данные: Примеры тем: «Объясни, что такое производная», «Напиши письмо преподавателю», «Составь список рекомендаций»

Ход работы:

1. Выберите 3 темы (см. выше).
2. Для каждой темы сформулируйте по 3 промпта:
   - В виде вопроса
   - В виде инструкции
   - В виде диалога (роль: преподаватель)
3. Сгенерируйте текст при следующих параметрах:
   - Температура: 0.2, 0.7, 1.0
   - Top_p: 1.0, 0.9
   - Max tokens: 50, 100
4. Оформите таблицу с результатами.
5. Проанализируйте:
   - Какие типы промптов дают наиболее точный результат?
   - Как влияет температура на вариативность?
   - Как изменяется стиль и длина при изменении max_tokens?

Пример промпта:

```
Ты — преподаватель по математике. Объясни студенту, что такое производная, простым языком.
```

Пример кода (опционально):

```
import openai
openai.api_key = 'your-key'
response = openai.ChatCompletion.create(
  model="gpt-3.5-turbo",
  messages=[{"role": "user", "content": "Объясни, что такое производная"}],
  temperature=0.7,
  top_p=1.0,
  max_tokens=100
)
print(response['choices'][0]['message']['content'])
```

Анализ результатов:

- Температура 0.2 даёт шаблонный, формальный ответ.
- Температура 1.0 — более креативный и иногда некорректный результат.
- Ролевая модель даёт более понятный и дружелюбный ответ.

Требования к отчёту:

- Таблица с параметрами и результатами генерации
- Примеры промптов (минимум 3 темы × 3 структуры)

- Аналитическая часть с выводами
- Скриншоты или вывод кода

Критерии оценки (зачтено/незачтено):
Зачтено: все типы промптов реализованы, минимум 6 генераций, таблица оформлена, выводы осмысленны.
Незачтено: отсутствуют 2+ варианта промпта, нет анализа, неполный отчёт.

Лабораторная работа №5: Создание пайплайнов взаимодействия с ИИ

Название: Разработка API-интерфейса генерации текста по промпту
Цель:
Реализовать простой сервис, отправляющий запрос к LLM через OpenAI API и возвращающий результат.
Задачи:

1. Настроить FastAPI-приложение с POST-запросом.
2. Отправить промпт на генерацию в модель OpenAI.
3. Обработать и вернуть ответ пользователю.

Ожидаемые результаты:

- Умение использовать API для генерации текста
- Навык работы с FastAPI и REST-интерфейсами
- Опыт логирования и форматирования вывода

Инструменты и библиотеки:

- Python 3.10+
- FastAPI
- OpenAI API
- Uvicorn, requests, dotenv

Ход работы:

1. Создайте файл `.env` с вашим API-ключом:

```
OPENAI_API_KEY=sk-...
```

2. Создайте файл `main.py`:

```
from fastapi import FastAPI, Request
import openai, os
from dotenv import load_dotenv
load_dotenv()

openai.api_key = os.getenv("OPENAI_API_KEY")

app = FastAPI()

@app.post("/generate")
async def generate(request: Request):
    data = await request.json()
    prompt = data.get("prompt")
    response = openai.Completion.create(
        engine="text-davinci-003",
        prompt=prompt,
        temperature=0.7,
        max_tokens=150
    )
    return {"result": response.choices[0].text.strip()}
```

3. Запустите сервер:

```
uvicorn main:app --reload
```

4. Протестируйте через Postman или curl:

```
curl -X POST http://127.0.0.1:8000/generate -H "Content-Type:
application/json" -d '{"prompt": "Напиши поздравление с Днем программиста"}'
```

Требования к отчёту: Скриншоты кода, Логика работы запроса, Примеры промптов и ответов, Описание структуры API и параметров

Критерии оценки: Зачтено: API запущен, протестирован, вывод получен, структура кода и отчета корректна. Незачтено: ошибки в коде, нет запуска, промпт не работает.

Проверяемые компетенции комплексом практических заданий: ML-2.1, LLM-5.1, LLM-5.2, ML-5.3

Зачетно-экзаменационные материалы для промежуточной аттестации (зачет)

Вопросы для подготовки к зачету

1. Что такое генеративный искусственный интеллект? Примеры его применения.
2. Как соотносятся задачи Data Science и задачи генеративного ИИ?
3. Объясните архитектуру трансформеров. Почему они эффективны для LLM?
4. Чем отличаются GPT и BERT по архитектуре и назначению?
5. Что такое Diffusion-модели и где они применяются?
6. Как работает температурный параметр в генерации текста?
7. Что такое top-p sampling и зачем он используется?
8. Какие параметры влияют на стиль и точность ответа LLM?
9. Почему важно учитывать контекст в промпт-инжиниринге?
10. Что такое zero-shot и когда он применяется?
11. Приведите пример few-shot промпта и опишите его преимущества.
12. В чём заключается метод Chain-of-Thought?
13. Объясните разницу между system prompt и user prompt.
14. Что такое пайплайн взаимодействия с LLM? Приведите пример.
15. Какие языки программирования используются для интеграции LLM?
16. Как можно реализовать генерацию текста через REST API?
17. Какие сложности возникают при встраивании промптов в пайплайн?
18. Какие бизнес-задачи можно автоматизировать с помощью LLM?
19. Приведите примеры шаблонов промптов для деловой переписки.
20. Как оценить ROI от внедрения генеративных моделей?
21. Что включает в себя формализация бизнес-задачи для ИИ?
22. Как можно измерить качество ответа LLM?
23. В чём смысл метрик BLEU, ROUGE и METEOR?
24. Что такое A/B тестирование промптов?
25. Как повысить надёжность генерации в нестабильной среде?
26. Что такое мультимодальный промпт?
27. Как работает генерация изображений по описанию?
28. Какие инструменты используются для мультимодальной генерации?
29. Какие проблемы возникают при обработке изображений и текста вместе?
30. Какие риски связаны с генерацией фейков и токсичных ответов?
31. Какие существуют методы фильтрации недопустимого контента?
32. Как учитывать юридические и этические нормы при работе с LLM?
33. Что такое ответственное ИИ и как оно связано с промпт-инжинирингом?
34. Какие этапы включает разработка эффективного промпта?
35. Как формализовать требования к проекту генеративного ассистента?
36. В чём отличие линейной и итеративной логики промпта?
37. Как документировать и версионировать промпты?
38. Какие навыки необходимы промпт-инженеру?
39. Какие инструменты облегчают работу с промптами?
40. Какие критерии используются для защиты курсовой работы по промпт-инжинирингу?

Перечень компетенций (части компетенции), проверяемых оценочным средством
LLM-5.1; LLM-5.2; LLM-5.4; ML-1.1; ML-2.1; ML-2.2; DL-2.1;

Практические задания к зачету

1. Промпты для автокомментирования кода

Задание: Напишите промпт, который получает функцию на Python и возвращает её подробный построчный комментарий. Протестируйте минимум 3 функции разной сложности.

2. Сравнение эффектов шаблонов генерации

Задание: Сравните качество ответов LLM на один и тот же вопрос в zero-shot, few-shot и chain-of-thought вариантах. Оцените точность, полноту и интерпретируемость.

3. Конструктор резюме

Задание: Реализуйте промпт-интерфейс, который получает ключевые данные о студенте (имя, опыт, образование) и генерирует готовое резюме. Вариативность: 2 языка и 3 стилистики (деловой, творческий, краткий).

4. Бизнес-кейс: генерация email

Задание: Сформируйте шаблон промпта, который получает информацию о клиенте и цели письма, и возвращает вежливое деловое письмо. Предусмотрите управление длиной и тоном.

5. LLM для обучения Python

Задание: Разработайте серию промптов-объяснений (не менее 5) для начинающих программистов. Темы: циклы, условия, функции, списки, рекурсия.

6. Тестирование параметров генерации

Задание: Один и тот же промпт выполните с разными параметрами temperature и top_p. Сравните результаты по разнообразию и адекватности вывода.

7. Мультимодальный генератор баннеров

Задание: Используя Stable Diffusion или DALL·E, создайте промпты для генерации рекламных баннеров по текстовому описанию продукта. Протестируйте минимум 3 случая.

8. Сценарии голосового ассистента

Задание: Напишите промпты, моделирующие диалоги голосового помощника в разных ролях: репетитор, медик, оператор доставки. Структурируйте промпты и ответы.

9. Промпт-интерфейс для генерации SQL

Задание: Составьте промпт, который преобразует текстовое описание запроса (например, «выбери все имена, где возраст больше 30») в корректный SQL-код. Протестируйте 5 кейсов.

10. Автоматический генератор викторин

Задание: Создайте промпт, который получает тему и количество вопросов, и генерирует викторину с 4 вариантами ответов и ключами. Оформите в формате JSON.

11. API-интерфейс генерации технических описаний

Задание: Напишите REST API на FastAPI, который получает название ИТ-продукта и генерирует маркетинговое или техническое описание через OpenAI API.

12. Реформаулировка и упрощение текста

Задание: Разработайте промпт, который получает научный абзац и переформулирует его «на пальцах» для школьника. Примеры: производная, логарифм, ядро матрицы.

13. Метапромпт: генератор промптов

Задание: Напишите промпт, который по краткому описанию задачи (например: «нужно сгенерировать SEO-заголовок») создаёт эффективный промпт для LLM.

14. Отладка и сравнение промптов

Задание: Сравните два промпта, решающих одну задачу (например, генерация обобщения статьи). Проведите анализ метрик качества (BLEU, ROUGE) и субъективной оценки.

15. Формализация этики взаимодействия

Задание: Составьте список тем, потенциально нарушающих правила безопасности (hate speech, вредные советы и т.д.). Разработайте безопасные альтернативные промпты.

16. Контекстный чат-бот по дисциплине

Задание: Сформируйте 10 промптов, покрывающих ключевые темы дисциплины (вопросы, определения, сравнения), и объедините их в логичный диалог с LLM.

17. Модель оценки эффективности промптов

Задание: Разработайте чек-лист оценки промптов (по критериям: краткость, точность, адаптивность, управляемость). Примените к 5 вашим промптам.

18. Анализ шаблонов генерации задач

Задание: Составьте промпт, который получает тему (например, производная) и генерирует учебную задачу с решением. Проведите оценку валидности и сложности.

19. Версионирование промптов

Задание: Реализуйте в Git или другой системе хранения файл с историей изменения промптов, комментариями и результатами генерации. Отразите влияние изменений.

20. Курсовой ассистент: чат по теме дисциплины

Задание: Создайте интерактивную систему (на основе Gradio, Streamlit или Telegram-бота), в которой промпты получают контекст из введённых пользователем данных. Поддержите темы Python, линейной алгебры и анализа.

Перечень компетенций (части компетенции), проверяемых оценочным средством
LLM-5.1; LLM-5.2; LLM-5.4; ML-1.1; ML-2.1; ML-2.2; DL-2.1;

4.2 Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций

Методические рекомендации, определяющие процедуры оценивания на зачете:

Процедура промежуточной аттестации проходит в соответствии с Положением о текущем контроле и промежуточной аттестации обучающихся ФГБОУ ВО «КубГУ».

Итоговой формой контроля сформированности компетенций у обучающихся по дисциплине является зачет. Студенты обязаны сдать зачет в соответствии с расписанием и учебным планом.

ФОС промежуточной аттестации состоит из вопросов к зачету и результатов текущего контроля.

Зачет по дисциплине преследует цель оценить работу студента за курс, получение теоретических знаний, их прочность, развитие творческого мышления, приобретение навыков самостоятельной работы, умение применять полученные знания для решения практических задач.

Форма проведения зачета: устно.

Результат сдачи зачета заноситься преподавателем в зачетную ведомость и зачетную книжку.

Оценивание уровня освоения дисциплины основывается на качестве выполнения студентом заданий текущего контроля и ответов на вопросы зачета.

Критерии оценки:

1. Оценка ответов на вопросы к зачету (50% итоговой оценки)

Зачет

Полные, развернутые ответы с демонстрацией глубокого понимания темы.
Ответы содержат основные идеи, но без углубленного анализа.
Использование примеров, формул, корректных терминов.
Возможны небольшие ошибки в деталях или формулировках.
Умение анализировать и сравнивать методы.

% выполнения: 60–100% (допускаются незначительные неточности).

Незачет

Отсутствие понимания ключевых концепций.
Грубые ошибки или неспособность ответить на большую часть вопросов.
% выполнения: <60%.

2. Оценка выполнения практических кейсов и лабораторных работ (50% итоговой оценки)

Зачет

Полное выполнение всех этапов кейса с инновационными решениями.
Достижение целевых метрик (например, F1 > 0.9).
Четкая документация кода и анализ результатов.
% выполнения: 60–100%.

или

Выполнены основные задачи, но без дополнительной оптимизации.
Незначительные отклонения от целевых метрик (например, F1 = 0.85).

или

Решены базовые задачи, но с критическими ошибками.
Низкое качество кода или отсутствие анализа.

Незачет

Невыполнение ключевых этапов.
Код нерабочий или отсутствует.
% выполнения: <60%.

Методические рекомендации, определяющие процедуры оценивания лабораторных работ:

Процедура оценивания лабораторных работ проходит в соответствии с Положением о текущем контроле и промежуточной аттестации обучающихся ФГБОУ ВО «КубГУ».

По каждой лабораторной работе оформляется отчет. Отчеты сдаются на проверку руководителю в течение курса по мере их выполнения, и защищаются студентами в установленном порядке.

При защите отчета студенту могут быть заданы вопросы и дополнительные задания по сути лабораторной работы, в том числе из списка контрольных вопросов к данной лабораторной работе. При неудовлетворительной оценке знаний студента по теме данного отчета, студент возвращается к повторному изучению соответствующих материалов, после чего допускается к повторной защите. Неудовлетворительно выполненный отчет также возвращается на доработку.

Отчет должен содержать заголовок, тему лабораторной работы, цель, задание, индивидуальную тему, описание хода выполнения работы, необходимые прикладные материалы (схемы, макеты документов и т.п.), в соответствии с требованиями к содержанию, и выводы по работе.

Оценочные средства для инвалидов и лиц с ограниченными возможностями здоровья выбираются с учетом их индивидуальных психофизических особенностей.

– при необходимости инвалидам и лицам с ограниченными возможностями здоровья предоставляется дополнительное время для подготовки ответа на зачете;

– при проведении процедуры оценивания результатов обучения инвалидов и лиц с ограниченными возможностями здоровья предусматривается использование технических средств, необходимых им в связи с их индивидуальными особенностями;

– при необходимости для обучающихся с ограниченными возможностями здоровья и инвалидов процедура оценивания результатов обучения по дисциплине может проводиться в несколько этапов.

Процедура оценивания результатов обучения инвалидов и лиц с ограниченными возможностями здоровья по дисциплине предусматривает предоставление информации в формах, адаптированных к ограничениям их здоровья и восприятия информации:

Для лиц с нарушениями зрения:

– в печатной форме увеличенным шрифтом,

– в форме электронного документа.

Для лиц с нарушениями слуха:

– в печатной форме,

– в форме электронного документа.

Для лиц с нарушениями опорно-двигательного аппарата:

– в печатной форме,

– в форме электронного документа.

Данный перечень может быть конкретизирован в зависимости от контингента обучающихся.

4.3. Методические указания по организации вычислительной инфраструктуры

Требования к аппаратному и программному обеспечению рабочих мест

Аппаратные требования. Для работы с инструментами промпт-инжиниринга студентам и преподавателю необходим стационарный компьютер или ноутбук (мобильные устройства не подходят) с современной конфигурацией. Рекомендуется многопроцессорный CPU (например, Intel Core i3/i5/i7 не ниже 4-го поколения или аналогичный AMD Ryzen) и оперативная память не менее 4 ГБ (рекомендуется 8 ГБ и выше). Компьютер должен быть оборудован стабильным подключением к сети Интернет (широкополосный доступ со скоростью не ниже 5–10 Мбит/с), что обеспечит быструю работу с веб-сервисами и API. Для преподавателя дополнительно желателен доступ к веб-камере и микрофону (для возможных онлайн-консультаций или прокторинга, если требуется), однако для основной работы по промпт-инжинирингу эти устройства не обязательны.

Программные требования. На рабочих станциях должна быть установлена современная операционная система: поддерживаются Windows 10 или 11, актуальные версии macOS, а также дистрибутивы GNU/Linux. Все системы должны регулярно обновляться для поддержания безопасности. Необходим современный веб-браузер (Google Chrome, Mozilla Firefox, Яндекс.Браузер, Safari, Opera – последняя стабильная версия). Браузер должен быть обновлен до актуальной версии на момент занятий для корректной работы веб-приложений LLM-платформ.

Дополнительное программное обеспечение. Студентам следует установить программные инструменты, необходимые для выполнения лабораторных работ и задач курса:

Git – система контроля версий для работы с репозиторием (установить последнюю версию клиента Git).

Инструменты для разработки – удобный текстовый редактор или IDE (Visual Studio Code, PyCharm и др.) для написания и редактирования промптов и кода.

Python и библиотеки – интерпретатор Python (рекомендуется версия 3.10+) и менеджер пакетов pip/conda, поскольку многие автотесты и скрипты могут быть написаны на Python. При необходимости – установить Jupyter Notebook/Lab для выполнения ноутбуков или использовать онлайн-среды (Google Colab и др.), как указано в материалах курса.

Docker – платформу контейнеризации необходимо установить на рабочем месте преподавателя (и при возможности у студентов) для локального запуска контейнеров, если планируется тестирование или развёртывание в контейнеризированной среде. Преподавателю Docker потребуется для настройки собственного Runner'а CI или проверки решений в идентичном контейнере локально. Примечание: на Windows Docker Desktop требует Windows 10/11 Pro или активации WSL2. На Linux и macOS Docker поддерживается напрямую.

Браузерные расширения/интеграции LLM – по желанию, можно установить расширения для взаимодействия с LLM, например, плагины для VS Code (CodeGPT и аналогичные) или Jupyter, однако это не является строгим требованием.

Все программное обеспечение должно быть настроено таким образом, чтобы студенты имели доступ ко всем необходимым инструментам во время занятий, а преподаватель обладал средствами администрирования инфраструктуры (доступ к репозиториям, управлению CI/CD, результатам тестов и пр.). Убедитесь, что на всех машинах отключены лишние ограничения доступа (например, брандмауэр пропускает соединения по HTTPS на внешние сервисы LLM, порты 80 и 443 открыты для исходящих подключений). Также необходимо наличие прав установки ПО на машинах или взаимодействие с системным администратором для установки требуемых программ.

Архитектура вычислительной инфраструктуры. Общий подход. В рамках курса организована современная вычислительная инфраструктура, обеспечивающая совместную работу над заданиями, автоматизированное тестирование решений и подключение к сервисам ИИ. Основу инфраструктуры составляют: система контроля версий с удалённым репозиторием (GitLab), конвейер непрерывной интеграции/развертывания (CI/CD) с использованием GitLab CI (Runner и Docker-контейнеры), набор автотестов для проверки решений студентов и шаблонные репозитории (template repositories) для упрощения начала работы над заданиями.

Система контроля версий (GitLab). Для хранения и версионирования кода и промптов каждого студента используется Git-репозиторий на платформе GitLab. Преподаватель создает шаблонный репозиторий – эталонную структуру проекта, в котором содержатся базовые директории, файлы заданий, примеры и конфигурация CI. Этот шаблон включает необходимую структуру папок для выполнения домашних заданий и механизм их сдачи на проверку (грейдер). Для каждого практического задания студент либо форкает шаблонный репозиторий, либо получает индивидуальный репозиторий, созданный на основе шаблона (GitLab поддерживает создание проекта из Custom Project Template, что ускоряет развёртывание одинаковых структур для всех студентов). В репозитории студента хранится его код решений: промпты, скрипты автоматизации, отчёты в формате Markdown/Notebook и прочие артефакты, необходимые по заданию.

Преподаватель и ассистенты подключаются к соответствующей группе проектов в GitLab, имея права maintainer/owner для мониторинга активности, ревью кода и управления настройками. История коммитов позволяет отслеживать прогресс студента и соблюдение сроков сдачи. Аппаратные требования для GitLab-сервера: допускается использовать либо внешний сервис (например, GitLab.com с приватными репозиториями или университетский сервер GitLab) – выбор зависит от требований безопасности. Если используется облачный GitLab, убедитесь, что у студентов есть доступ к интернету и сервис не заблокирован. Репозитории должны быть приватными (чтобы работы студентов не были общедоступны) и настроены таким образом, чтобы студенты не могли просматривать решения друг друга (в GitLab можно ограничить доступ к проектам внутри группы).

CI/CD конвейер (GitLab Runner и Docker). Для автоматизации тестирования и развёртывания используется встроенная система GitLab CI/CD. В каждом репозитории присутствует файл конфигурации конвейера `.gitlab-ci.yml`, описывающий этапы (stages) и задания (jobs) для проверки решений. Например, конвейер может состоять из этапов: test (запуск автотестов), report (сбор и публикация отчетов) и deploy (необязательный этап, например, для деплоймента результатов или файлов отчета на Pages).

GitLab Runner – исполнитель задач CI – настроен для данного курса. Может быть использован shared runner (облачный, предоставляемый GitLab SaaS) или условно выделенный раннер, развернутый преподавателем на отдельной машине или сервере кафедры. Раннер должен быть сконфигурирован с использованием Docker-экзекутора, что позволяет запускать тесты в изолированных контейнерах с заранее заданным окружением. Например, для тестирования Python-скриптов можно использовать Docker-образ на базе Python 3.10 с предустановленными зависимостями (numpy, pytest, etc.). Контейнер обеспечивает единообразие среды: таким образом, в любом месте все тесты запускаются в идентичной конфигурации, исключая проблему "работает на моей машине".

Примечание: Если используется собственный GitLab Runner, на машину раннера необходимо установить Docker и зарегистрировать Runner токеном проекта/группы. Раннер должен иметь доступ к интернету, чтобы при необходимости скачивать зависимости и обращаться к API LLM (подконтрольно, см. раздел безопасности). В конфигурации Runner рекомендуется ограничить максимальное время выполнения job и количество параллельных job согласно возможностям оборудования (например, не более 2 concurrent jobs на 4-ядерном CPU с 8 ГБ RAM).

Автоматическое тестирование (автотесты). В шаблонные репозитории включены наборы автотестов, реализованные на скриптах (например, на Python с использованием PyTest или Unittest) либо других языках, в зависимости от специфики задания. Автотесты проверяют корректность выполнения заданий: например, для задач по проектированию промптов автотест может вызывать функцию, генерирующую ответ LLM на фиксированный запрос, и сравнивать с эталонным ключевым словом; для задач программирования – запускать юнит-тесты к сгенерированному коду.

При каждом коммите или merge request репозиторий запускается конвейер CI: в Docker-контейнере выполняются автотесты. Если все тесты проходят успешно, конвейер помечается как “зелёный” (прошёл), иначе – “красный” (найдены ошибки). Результаты тестирования сохраняются в виде отчётов. GitLab CI умеет парсить отчёты в формате JUnit/XML и отображать сводку прямо в интерфейсе (вкладка Test Reports в pipeline), где видно число пройденных/упавших тестов. Это позволяет студенту и преподавателю быстро оценить результаты автоматической проверки.

Шаблонные репозитории и примеры. Шаблонный репозиторий содержит не только структуру проекта, но и пример конфигурации `.gitlab-ci.yml` и пример базовых тестов. Это методически важно: студенты с первого задания привыкают к структурированию своего решения и к прохождению автопроверки. Например, в лабораторной работе №1 студенту предлагается сформулировать 5 промптов и провести их генерацию через разные LLM-платформы. В шаблоне для этой работы может быть заготовлен ноутбук Jupyter с каркасом для эксперимента (ячейки для запросов к DeepSeek, ChatGPT и др.) и автотест, который проверяет наличие заполненных полей (например, действительно ли студент внёс 5 промптов и получил ответы от всех требуемых моделей). Шаблонные репозитории облегчают запуск проекта, снижают количество настроечных ошибок и обеспечивают единообразие при проверке (структура решений у всех студентов сходна, что упрощает автоматизированный анализ).

Рекомендации по выбору и подключению к LLM-платформам

Обзор используемых LLM-платформ

GigaChat (Сбербанк)

Доступность в России: GigaChat – отечественная LLM-платформа от Сбера – полностью доступна российским пользователям. Сервис был запущен весной 2023 г. как русскоязычный аналог ChatGPT. К октябрю 2023 г. GigaChat стал доступен через веб-интерфейс, Telegram-бот и соцсеть VK. Для использования не нужен VPN; авторизация происходит через Sber ID или социальные сети. Таким образом, студенты из РФ могут легко воспользоваться GigaChat. Кроме того, бизнес-версия через облачное API была запущена Сбером в режиме бета в 2024 г. и к 2025 г. стала широко доступна. Например, на портале SberCloud есть GigaChat API с документацией и возможностью получения ключа. Важно: так как платформа отечественная, проблем с санкциями или платежами нет – оплата в рублях, поддержка российских карт. GigaChat является одним из рекомендуемых инструментов для курса ввиду легкой доступности и адаптации под русский язык.

API и интеграция: GigaChat предоставляет API для разработчиков (через Sber Developer Portal). Интерфейс API – RESTful, с аутентификацией по API-ключу, позволяет интегрировать GigaChat во внешние приложения. Формат взаимодействия описан в документации: запрос содержит поле с текстом пользователя, а ответ – сгенерированный моделью текст. Поддерживаются также отдельные методы для создания диалоговых агентов (в документации упоминается использование LangChain для построения агентов с GigaChat). Таким образом, через API можно вести сессии диалога, отправлять новые сообщения с указанием идентификатора сессии, получая контекстно связанные ответы. GigaChat API интегрирован в экосистему Сбера, но его может использовать любой сторонний разработчик (достаточно зарегистрироваться на developers.sber.ru). Для учебных проектов это значит, что можно, например, написать тестовый чатбот, подключив к нему GigaChat как мозг. Важно отметить: UI GigaChat (веб и боты) – тоже доступны, и для простых заданий достаточно просто пользоваться ими, не кодируя.

Настройка параметров генерации: Через публичный API GigaChat доступны некоторые служебные параметры (в документации упоминается раздел B2.4 «Служебные параметры» на SberUniversity). Вероятно, можно регулировать максимум символов ответа, стиль (например, режим «строгий» или «креативный»). Однако подробностей в открытом доступе мало. Судя по реализованному функционалу, GigaChat имеет режимы: «Lite» – более быстрая модель, «Pro»/«Max» – более продвинутые, тяжелые модели. Пользователь может выбирать модель (в интерфейсе бота, например, была команда переключения версии). Но явного изменения temperature или top_p нет в пользовательском интерфейсе. Можно предположить, что внутренне GigaChat Pro работает с оптимизированными параметрами для информативных ответов (менее «галлюцинирует»). В целом, платформы Сбера ориентированы на простоту: настроек генерации немного. Тем не менее, для исследовательских целей сотрудники Сбера сами используют LangChain с GigaChat, а там обычно подразумевается возможность указания stop sequences, memory window и др. Итого: ограниченная конфигурируемость – обычному пользователю достаточно получать качественный текст, а тонкие настройки доступны разработчикам при интеграции.

Отслеживание запросов и лимиты: GigaChat имеет тарифные планы: на 2023–2024 был бесплатный доступ с ограничением по символам и возможному числу запросов в час (конкретные цифры не раскрывались публично). В 2024 г. Сбер представил обновление GigaChat 2.0 с версиями Pro и Max. GigaChat Max стал доступен всем желающим бесплатно (через веб, Телеграм), но в режиме раннего доступа API, вероятно, были ограничения по

количеству запросов. К началу 2025 г. публичный API GigaChat заработал для бизнеса с тарификацией (на SberCloud), однако для образования есть возможность сотрудничества через СберУчебу. В новостях не раз отмечалось, что контекст у GigaChat был увеличен до 32 тыс. токенов – это подтверждается и Habr-статьей Сбера. То есть сейчас GigaChat Pro/Lite поддерживают до ~60 страниц текста в одном запросе. Это отличный показатель (на уровне GPT-4 32k и Claude 100k). Лимит длины ответа, вероятно, также большой – модель может генерировать пространные тексты. Sber заявляет об устранении «очередей» для премиум-пользователей: веб-версия GigaChat Max выдаёт ответы практически сразу, без ожидания. Для отслеживания использования API СберCloud предоставляет метрики и биллинг в личном кабинете. В учебном же формате можно особо не беспокоиться – если даже все студенты одновременно начнут диалог, GigaChat выдержит (Сбер обладает мощными вычислительными ресурсами). В случае появления коммерческих лимитов, вуз может запросить у Сбера льготный квотированный доступ.

Интеграция и CI/CD: GigaChat легко интегрируется в приложения, особенно нацелен он на бизнес-процессы. Например, существуют готовые модули для 1С и CPM с подключением GigaChat. В рамках CI/CD курса промпт-инжиниринга GigaChat API можно использовать для автоматизированного тестирования на русском языке. Его преимущество – стабильность ответов на русском: модель обучена на русскоязычных данных, и, как отмечают, GigaChat Pro по ряду задач уступает лишь топовым зарубежным моделям. Преподаватель может написать скрипт, который прогоняет набор русских промптов через GigaChat и проверяет наличие ключевых слов в ответе (например, в задаче на код – что модель сгенерировала def function():). Поскольку GigaChat находится под управлением Сбера, изменения модели происходят планово (от версии 1.0 к 2.0 и т.д.), но не внезапно каждый день. Поэтому тесты на одной версии будут действительны хотя бы несколько месяцев. В CI можно также настроить два варианта GigaChat (Lite и Max) для сравнения. Сберуниверситет публиковал учебные материалы, где описывается, как создавать агентов и навыки с GigaChat API – этим можно воспользоваться при составлении продвинутых лабораторных.

Мультимодальность: GigaChat эволюционировал в мультимодальную модель. Изначально он отвечал только на текстовые запросы. В 2024 г. была обучена GigaChat Vision – модель понимания изображений. В октябре 2024 г. Сбер объявил, что GigaChat Pro может принимать картинки в качестве части запроса и использовать их как контекст. То есть студент может загрузить фотографию, и GigaChat опишет, что на ней (например, посчитает людей, оценит их одежду, даст совет по стилю). Модель научилась также распознавать печатный и рукописный текст на изображениях, формулы, таблицы, графики – фактически встроена OCR и базовый анализ данных. Например, можно сфотографировать страницу учебника, и GigaChat выдаст краткое содержание и ключевые тезисы. Это крайне полезно для образования (конспекты, уравнения). Кроме того, GigaChat получил функцию генерации и редактирования изображений: встроена модель «Мальвина» для локального редактирования фото по сегментам. Пользователь может загрузить картинку и текстово описать, что на ней изменить – GigaChat внесёт правки (пример: поменять фон на фото, перерисовать объект и т.п.). Такая точечная генерация уникальна среди LLM-платформ. Также, по некоторым данным, GigaChat умеет генерировать новые изображения по описанию (возможно, через связку с Malvina или Kandinsky 2.1). И наконец, Сбер научил GigaChat «слышать»: в 2025 вышла статья о том, как добавили обработку аудио (речь и звуки) для нативного понимания аудиоинформации. Вероятно, это означает, что GigaChat может транскрибировать аудиофайлы или отвечать голосом. В пресс-релизах отдельно не анонсировано, но технически модальность аудио упоминается. Подытожим: GigaChat –

одна из самых мультимодальных платформ наравне с Gemini и Grok. Он понимает и текст, и изображения, и (в перспективе) аудио, а отвечает текстом или измененными изображениями. Для курса это открывает большие возможности: можно дать студентам задания с анализом картинок (инфографика, диаграммы), и они смогут использовать GigaChat для получения ответов.

Условия доступа и тарифы: На данный момент использование GigaChat для физических лиц бесплатно – ни веб, ни бот не требуют оплаты. Сбер планирует монетизировать модель через бизнес-API: уже сейчас на SberCloud указан тариф (например, GigaChat Lite бесплатно, GigaChat Max за отдельную плату, детали на портале). Также упоминалось о подписке для расширенных функций, но в открытом доступе модель работает без ограничений. Например, обновленный GigaChat Max открыт всем желающим. Для курса это значит, что студенты могут свободно тренироваться с моделью и выполнять задания, не думая об истощении лимита. При массовом использовании в аудитории может проявиться очередь на ответ (если сервер перегружен), но Сбер масштабирует мощности. Если вуз захочет интегрировать GigaChat в LMS или сервис, можно обратиться к Сберу за партнерским доступом (что они, возможно, поддерживают для образовательных учреждений). Академических лицензий как таковых нет – они не нужны, доступ и так открыт.

Этические и правовые вопросы: GigaChat, являясь российской моделью, следует законодательству РФ. Это значит, что политически чувствительный или запрещенный законом контент фильтруется. Например, Сбер не допустит экстремистских высказываний, пропаганды наркотиков и пр. В Википедии упоминалось, что у ЯндексGPT цензурируется тема Степана Бандеры – можно ожидать похожего и от GigaChat (склонен избегать острых углов по Украине и т.п.). Стиль ответов GigaChat довольно нейтральный, возможно даже суховатый (ориентирован на деловой русский язык). В последнем обновлении GigaChat Pro научился лучше форматировать текст (заголовки, списки), что полезно, но может и приукрасить ответ лишними деталями. Этически GigaChat старается быть корректным, хотя сообщество отмечало, что ранние версии иногда генерировали недостоверные факты наукообразным языком. Сбер пытается этого избежать: модель улучшалась на открытых бенчмарках (лидерборд MERA по пониманию русского языка, где GigaChat Max занял одно из первых мест). Это говорит о репутации: GigaChat стремится давать проверенные знания. Для студентов стоит подчеркнуть, что GigaChat – российская разработка, поэтому может иметь встроенные патриотические или лояльные нарративы (незначительные, но в сравнении с ChatGPT отличия в оценочных вопросах возможны). С точки зрения права, использование GigaChat на территории РФ безопасно – данные обрабатываются, скорее всего, в российском дата-центре, а не уходят за рубеж. Это плюс для работы с персональными или чувствительными данными: они остаются под юрисдикцией РФ. Однако все нормы академической честности также действуют – нельзя без ссылки вставлять сгенерированный GigaChat текст в работу студентов. Подытоживая, GigaChat – рекомендованная платформа для русскоязычных заданий, при условии понимания ее слегка официального тона и цензурных ограничений российского законодательства.

YandexGPT (YaGPT, Яндекс)

Доступность в России: Модель YandexGPT изначально создана для российского рынка и интегрирована в множество сервисов Яндекса. С мая 2023 г. Яндекс открыл доступ к YandexGPT в своем ассистенте Алиса, затем встроил в Поиск, Маркет, Браузер и др. продукты. Для пользователей это проявляется, например, в том, что на главной странице Яндекса есть чат для вопросов с ИИ. Доступ свободный и бесплатный, нужен лишь Яндекс.ID. VPN не требуется, сервисы Яндекса внутри страны прекрасно работают. К

концу 2024 г. Яндекс сообщил, что 800 компаний участвовали в тестировании бизнес-версии YandexGPT. В ноябре 2024 г. запущен инструмент AI Assistant API на базе YandexGPT, то есть API доступен через Yandex Cloud для организаций. Следовательно, для нужд курса YandexGPT тоже вполне доступен: студенты могут использовать его на портале Яндекса (в Алисе или в Поиске), а преподаватели – при желании подключить API через облако (потребуется аккаунт на Yandex Cloud).

API и формат взаимодействия: Яндекс предоставляет YandexGPT API в составе Yandex Cloud (раздел Machine Learning). Доступ возможен в двух режимах: через API (REST, синхронный или асинхронный) и через Playground (веб-консоль для тестирования моделей). Для API требуется получить токен, после чего можно отправлять POST-запросы с текстом. Примечательно, что Яндекс предлагает две версии модели для бизнеса: одна – асинхронная и более «умная» (для сложных задач, но отвечает с задержкой), вторая – быстрая для real-time чатов. Такой подход позволяет в коде выбирать: либо ждать более качественного развернутого ответа, либо получить краткий сразу. Формат ввода – обычный текст (вопрос или реплика диалога), опционально с параметрами (какую версию модели использовать). Вывод – сгенерированный текст ответа. Внутренне YandexGPT работает похоже на ChatGPT: запоминает контекст беседы (Алиса, например, поддерживает уточняющие вопросы). Однако через Cloud API, скорее всего, разработчик сам должен передавать историю, если нужен диалог. На практике для курса, вероятно, хватит пользовательского интерфейса (Алиса или чат на ya.ru). Но если потребуется автоматизация – YandexGPT API есть, хоть и требует регистрации в Яндекс Облаке.

Настройка генерации: В публичных сервисах Яндекса (Поиск с ИИ, Алиса) у пользователя нет настроек – ответы генерируются по внутренним параметрам, которые Яндекс подбирает для оптимального качества. Яндекс явно позиционирует YandexGPT как часть готовых решений (техподдержка, контент-генерация), поэтому предоставляет скорее конфигурацию на уровне «режимов использования», чем на уровне temperature. Например, в Yandex Cloud есть инструмент создания чатбота, где можно натренировать его на своих Q/A. Температуру, возможно, можно указать при вызове API (документация пока не в открытом доступе, но логично предположить, что какая-то креативность настраивается). Известно, что YandexGPT версии 5 (февраль 2025) используется в Алисе и поддерживает голосовой диалог и генерацию изображений. Значит, у Алисы есть переключение между голосовым и текстовым режимом. Но глубоко на уровне API – конфигурирование ограничено. Для наших целей: YandexGPT конфигурируется через “сценарии” – можно обучить на своем датасете, задать тон (вежливый/неформальный). А такие параметры, как длина ответа, наверняка можно задать (Алиса, например, по умолчанию не делает очень длинных монологов). В итоге, YandexGPT более “стабильно настроена” модель, чем “игрушка для экспериментов” – что хорошо для надежности, но оставляет меньше простора для творчества, чем Llama или ChatGPT API.

Отслеживание запросов и лимиты: Для конечных пользователей YandexGPT почти прозрачна – например, Алиса может ответить на сотни вопросов, лишь иногда скажет: “давай по очереди” (это скорее усталость голосового ассистента). В Поиске Яндекса при серии сложных запросов ИИ-чат может временно отключиться, чтобы не перегружать – но четких ограничителей не афишируется. Для API через Yandex Cloud действуют стандартные квоты: на этапе бета 2023 г. компания вручную распределяла доступ (800 компаний-участников), в 2024 г. открыли шире. Бизнесу предлагаются 2 версии модели, как сказано, – видимо, их capacity тоже различна (async модель может потребовать больше времени, но справиться с большим объемом). Конкретный лимит контекста Яндекс не раскрывал, но косвенно: в октябре 2023 Яндекс заявлял, что модель умеет пересказывать

видеоролики до 4 часов, а тексты – до 30 тыс. знаков (~≈5000 слов), что ~5000 токенов. Версия 5, вероятно, увеличила контекст. Пока можно ориентироваться на ~8k токенов. Ограничения по токенам на ответ – есть ли? Алиса обычно отвечает не слишком длинно, возможно, стоит предел. В API же разработчик может запросить продолжение до N символов. Мониторинг использования API доступен через метрики Yandex Cloud (считаются запросы и, вероятно, объем данных). Оплата у Яндекс идет за количество символов или за время генерации (нужно уточнять тарифы, но обычно – помегабайтно или по 1000 токенов, аналогично AWS/GCP). В рамках учебного тестирования, при наличии бесплатного кредица на Яндекс Облаке, можно выполнить достаточное число запросов.

Интеграция в CI/CD: YandexGPT API можно встроить в решения поддержки клиентов, контент-сервисы и т.п. – этим занимаются в бизнесе. В нашем случае CI/CD можно задействовать YandexGPT для тестов на русском языке с корпоративным стилем. Например, если мы учим студентов настраивать AI для техподдержки, YandexGPT – подходящая модель (учитывает форму общения, умеет цитировать документы). Разработчики могут использовать Playground в Yandex Cloud для проверки промптов – это аналог Postman'а для ИИ. В CI, чтобы писать автотесты, нужно иметь платную подписку (или бесплатный пробный баланс).

Мультимодальность: YandexGPT 5 – мультимодальная система. Она работает в Алисе, отвечая голосом (то есть синтез речи – это не свойство языковой модели, а связка со SpeechKit, но для пользователя выглядит цельно). Также, начиная с версии 2, Яндекс научил модель генерировать изображения (в сервисе Шедеврум, июнь 2023). Пользователи могут попросить сгенерировать картинку, и YandexGPT предложит иллюстрацию к тексту. Более того, YandexGPT применил себя в переводе: улучшил машинный перевод видео, суммируя речь. То есть связка YandexGPT + speech-to-text позволяет на лету обобщать аудиоматериал.

Что касается ввода изображений или аудио: таких пользовательских функций пока у Яндекса нет (в Алисе нельзя отправить фото для анализа, во всяком случае пока). Но Яндекс активно работает над мультимодальностью – вероятно, внутренние версии модели умеют и это. В YandexGPT 5 Pro (на которой работает Алиса) заявлено, что она генерирует тексты и изображения – возможно, по сложному запросу Алиса сама придумает картинку и покажет. Однако на практике это не так популярно, как у ChatGPT или GigaChat, поэтому упомянем как факт: мультимодальный вывод (текст+картинка) – да, мультимодальный ввод (картинка→текст) – нет, аудио – только через обвязку (распознавание и озвучка вне LLM).

Для курса полезно то, что YandexGPT может взаимодействовать с другими технологиями Яндекса – например, можно задействовать API зрительного распознавания (Vision) и затем отправить описание в YandexGPT. Яндекс анонсировал также Gemini Apps (не путать с Google Gemini) – решение для автоматизации рутин с помощью LLM в Google Workspace аналогах. Пока основное – текст.

Условия доступа: Для частных лиц YandexGPT представлен через бесплатные продукты (никакой прямой оплаты за ответы Алисы не берется). Бизнес-доступ идет через Yandex Cloud – там, конечно, услуги платные (но есть бесплатный пробный период). Для вузов Яндекс нередко предлагает гранты или бесплатные квоты на облачные сервисы – можно уточнить у регионального представителя. Но даже без этого можно много чего сделать на бесплатных инструментах: например, веб-версия Алисы способна решать задачи, и студенты могут ею пользоваться как альтернативой ChatGPT. Алиса доступна и как мобильное приложение, и через браузер.

Отдельно стоит упомянуть: Яндекс в октябре 2023 открыл API YandexGPT в Казахстане (для локальных разработчиков), что намекает на стремление продвигать модель в СНГ. Может быть, в РФ API тоже станет свободнее и с trial-квотой. Академической

лицензии нет – если нужно много запросов, возможно заключение соглашения на облачные кредиты.

Этические и юридические ограничения: YandexGPT находится под контролем модерации Яндекса и соблюдает российские законы о информации. Известно, что Яндекс не позволит генерировать незаконный контент и кое-что цензурирует – например, вопросы про украинского националиста могут быть пропущены или дан уклончивый ответ. Модель обучалась на русскоязычных данных (книги, СМИ, интернет), поэтому несёт в себе определенные культурные предвзятости. Критики отмечали, что большие интернет-компании, включая Яндекс, осторожны в острых темах – YandexGPT, вероятно, не скажет ничего резко противоправного, но может и не удовлетворить пользователя, если тема политически заряжена. В образовательном плане это не критично, но упомянуть нужно: для заданий, связанных с политикой или историей, ответы Яндекса могут оказаться более скудными или шаблонными. С другой стороны, YandexGPT проявил себя хорошо в прикладных областях – перевод, справка, код.

Что касается точности: Яндекс стремится минимизировать “галлюцинации”. Например, YandexGPT интегрирован в Поиск, где отвечает с указанием источников (аналог Perplexity) при включенной опции быстрого ответа. Это повышает релевантность. Яндекс также анонсировал открытие исходных версий модели Gemma (2B и 7B параметров) в 2024 г. как ответ на Meta – то есть, признает важность открытости. Впрочем, пока большая YandexGPT закрыта. Юридически, используя YandexGPT, мы подчиняемся пользовательскому соглашению Яндекса. Оно требует не вводить нелегальный контент, не злоупотреблять сервисом. Яндекс явно пишет, что модель может ошибаться и фантазировать – преподавателю стоит предупредить об этом студентов.

Персональные данные: если студенты вводят какие-то свои данные, они уходят на сервера Яндекса (расположены в РФ), где могут логироваться для улучшения моделей. В 2023 г. Яндекс заявлял, что будет обучать модель на обезличенных данных пользователей – то есть chat-диалоги, скорее всего, используются. Поэтому не стоит вводить в Яндекс секретную информацию без необходимости. В остальном, YandexGPT – безопасный и надежный инструмент в рамках российского правового поля, и его стоит использовать в курсе наряду с GigaChat, чтобы студенты увидели возможности отечественного ИИ.

ChatGPT (OpenAI)

Доступность в России: ChatGPT официально недоступен для пользователей из России – OpenAI блокирует российские IP и оплату через российские карты. Для доступа необходим VPN; подписку ChatGPT Plus (доступ к GPT-4) можно оплатить только зарубежной картой или альтернативными способами. Это создает определенные сложности при использовании ChatGPT в России.

API и формат взаимодействия: Имеется API (OpenAI Chat Completions), позволяющий отправлять диалоги в формате ролей (system, user, assistant) и получать ответ модели. Формат аналогичен структуре беседы – список сообщений с указанием роли и контента. Через API доступны модели GPT-3.5 и GPT-4; ввод и вывод – текст (UTF-8). В веб-интерфейсе ChatGPT диалог ведется аналогично, но без возможности явного задания параметров генерации.

Настройка параметров генерации: В OpenAI API можно настраивать temperature (степень творчества), max_tokens (максимальная длина ответа), top_p и другие параметры для управления стилем и детальностью ответа. В веб-версии ChatGPT такие настройки напрямую недоступны, но частично поведение модели регулируется скрытой системной подсказкой.

Отслеживание запросов и ограничения: OpenAI предоставляет разработчикам дашборд для мониторинга количества токенов и запросов. Имеются лимиты контекста: GPT-3.5 поддерживает 4 тыс. токенов, GPT-4 – 8 тыс. (в версии Plus) или 32 тыс. токенов в расширенной версии. В веб-версии ранее действовали ограничения на число сообщений в час, тогда как у Claude динамические лимиты (зависят от нагрузки). Пользователи ChatGPT Plus могут просматривать историю запросов и расход токенов через веб-интерфейс.

Интеграция в CI/CD: Возможно подключение ChatGPT API в автоматизированные сценарии (например, для тестирования промптов). При интеграции следует учитывать недетерминированность генерации – для воспроизводимости результатов можно фиксировать temperature=0. Также важно заложить обработку сетевых задержек и ограничений скорости. Некоторые организации используют юнит-тесты промптов, проверяя, что обновления модели не нарушают целевого поведения ответа.

Мультимодальность: В последние обновления ChatGPT получил ограниченную мультимодальность. Модель GPT-4 умеет анализировать изображения – пользователь может прикрепить картинку и задать вопрос по ней. Это доступно подписчикам Plus: в течение 2023 г. OpenAI внедрила поддержку распознавания изображений и голосового общения. Через мобильное приложение ChatGPT можно общаться голосом (нейросеть распознаёт речь и отвечает синтезированным голосом). Таким образом, ChatGPT Plus «видит, слышит и говорит», однако в API эти функции пока ограничены (изображения принимаются только в интерфейсе ChatGPT, а не через REST API). В целом ChatGPT – текстово-ориентированная платформа с растущими мультимодальными возможностями.

Условия доступа: Базовая версия (GPT-3.5) доступна бесплатно (через VPN), но с ограничениями по скорости и контексту. Полноценный доступ к GPT-4 и мультимодальным функциям предоставляется по платной подписке ChatGPT Plus (≈20 USD/мес, недоступна для оплаты из РФ без обходных методов). Для академических целей OpenAI отдельные бесплатные лицензии не предоставляет, однако существуют образовательные программы и гранты от Microsoft/OpenAI (Azure OpenAI) для вузов – их можно рассмотреть при необходимости. В целом, для студентов в РФ ChatGPT – менее доступная платформа по сравнению с локальными или открытыми решениями.

DeepSeek (Китай)

Доступность в России: Китайская платформа DeepSeek не блокируется для России – доступ возможен без VPN, сервис стабильно работает на территории РФ. В 2025 году DeepSeek обрела большую популярность: ее мобильное приложение стало лидером загрузок, обойдя ChatGPT. Для регистрации и оплаты могут потребоваться нестандартные методы (например, аккаунт AliPay), но сам сервис не ограничивает российских пользователей. По отзывам, DeepSeek – одна из немногих передовых LLM, доступных из РФ без сложных обходов.

API и формат взаимодействия: DeepSeek предоставляет полноценную платформу для разработчиков (DeepSeek API Platform). API совместим с библиотеками OpenAI – запросы формируются схожим образом. Например, для Python можно использовать клиент OpenAI с изменением base_url на https://api.deepseek.com и отправлять chat.completions запросы с ролью system и сообщениями. Такой дизайн упрощает интеграцию DeepSeek: разработчики, знакомые с OpenAI API, могут легко переключиться на DeepSeek. Ввод/вывод – многострочный текст (чат-режим), также есть специализированные модели (код, математическая, визуальная и др.). DeepSeek поддерживает российский и английский языки, а также китайский (но основной упор – на китайский и английский рынки).

Настройка генерации: Через API DeepSeek можно управлять параметрами генерации, аналогично OpenAI. Поддерживаются temperature, max_tokens и потоковая

выдача ответов (streaming). В бесплатном веб-интерфейсе таких настроек нет – он рассчитан на максимальное удобство. Однако продвинутые пользователи через API могут тонко настраивать стиль вывода. В целом логика работы DeepSeek «максимально похожа» на другие нейросети, то есть привычные параметры налицо.

Отслеживание запросов и лимиты: DeepSeek ввела систему кредитов и тарифов к 2025 году. Бесплатный тариф предоставляет ограниченное число запросов и урезанные функции (например, ограниченная генерация изображений и анализ файлов). Платные подписки (Pro за $8/мес) дают приоритет к ресурсам и снятие ограничений на число обращений. Для API существует модель оплаты per token: стандартная цена $0.27 за 1M входных токенов (кэш пропущен) и $1.10 за 1M выходных токенов, причем в непиковые часы действует скидка 50%. Таким образом, стоимость генерации у DeepSeek значительно ниже, чем у GPT-4 (для сравнения, у OpenAI GPT-4 порядка $30 за 1M токенов вывода). Глубина контекста в модели DeepSeek-Chat – до 8 тыс. токенов вывода (полный контекст ~16k). На конец 2024 г. сообщения о максимально поддерживаемом контексте до 32k токенов отсутствуют, но новые версии (например, DeepSeek-R1) могут увеличить это значение. Отслеживать использование API можно через личный кабинет разработчика; там же пополняется баланс токенов.

Инструменты интеграции и CI/CD: DeepSeek API уже поддерживается популярными фреймворками для LLM (например, LangChain). SberDevices в одном из примеров отмечает, что LangChain нативно умеет работать с GigaChat API, аналогично сообществом добавлена поддержка DeepSeek API. Это упрощает создание pipeline для тестирования промптов – можно быстро переключать вызовы между OpenAI и DeepSeek. При интеграции в CI/CD DeepSeek выгоден отсутствием необходимости VPN и стабильной работой из РФ. Необходимо лишь предусмотреть хранение API-ключа и пополнение кредитов. Для образовательных задач DeepSeek может быть встроен, к примеру, в веб-приложение для студентов без риска блокировки соединения.

Мультимодальность: Платформа DeepSeek изначально текстовая, но стремительно развивается к мультимодальности. По состоянию на 2025 г. доступны специализированные модели: DeepSeek-VL – экспериментальная модель для работы с изображениями, и DeepSeek-Chat – основная универсальная модель с поддержкой простых визуальных функций. Бесплатный тариф допускает лишь ограниченную генерацию изображений и анализ вложенных файлов, тогда как в платном доступен полный функционал: можно прикреплять изображения, таблицы, PDF-файлы для анализа. DeepSeek-VL предназначена для профессиональной работы с изображениями и обладает «кардинально расширенными визуальными возможностями» – например, она может описать содержимое картинки или сгенерировать новое изображение по запросу. Информация о поддержке аудио отсутствует, основной упор – текст + изображения. Таким образом, DeepSeek уже мультимодальна (текст и изображения) в рамках подписки Pro, что делает ее конкурентом GPT-4 Vision.

Условия доступа: DeepSeek предлагает бесплатный базовый план (с существенными ограничениями по функционалу). Полноценная работа требует подписки Pro ($8/месяц), которая дает неограниченное число запросов к флагманской модели V3 и приоритет при нагрузках. Для активных пользователей/API предусмотрен тариф по токенам, а для организаций – Enterprise-решения. Оплата из России затруднена: карты UnionPay и обычные не всегда привязываются. Пришлось использовать AliPay с юаневым кошельком – это указывает, что официально DeepSeek не работает с российскими платежными системами, но через китайские можно заплатить. Академических бесплатных лицензий у DeepSeek нет, однако сам факт наличия бесплатного веб-доступа (пусть и урезанного) дает возможность студентам познакомиться с платформой без оплаты.

Perplexity AI

Доступность в России: Perplexity (американская поисково-аналитическая LLM-система) доступна без VPN на территории РФ. Сервис работает через веб-интерфейс и мобильные приложения; регистрация не обязательна для бесплатного использования. Российские пользователи могут свободно задавать вопросы – ограничений по количеству запросов в бесплатной версии нет. Таким образом, Perplexity можно применять в курсе без технических барьеров, аналогично обычному веб-сайту.

API и формат взаимодействия: Изначально Perplexity ориентирован на конечных пользователей (вопрос-ответ с поиском), но в 2023–2024 гг. компания запустила официальный API для разработчиков. API позволяет делать HTTP-запросы к моделям Perplexity, формат схож с чат-API (принимаются сообщения или вопросы, можно получать ответы по мере готовности). Кроме того, Perplexity интегрирован с платформой OpenRouter, предоставляя унифицированный доступ наряду с моделями OpenAI. На практике Perplexity API работает через REST и совместим с клиентскими библиотеками OpenAI. Взаимодействие включает, помимо текста пользователя, дополнительные параметры (например, включение веб-поиска). Отличительная особенность Perplexity – ответ всегда сопровождается ссылками на источники информации, так как платформа осуществляет поиск по Интернету и цитирует найденные данные. Формат вывода – развернутый ответ + список URL-источников. В контексте обучения это ценно: студенты видят, откуда взялась информация, что повышает доверие и проверяемость результатов.

Настройка генерации: Пользовательский интерфейс Perplexity не предоставляет явных регуляторов вроде temperature – акцент на достоверности ответа. Однако в Pro-версии можно выбирать модель (например, более мощную GPT-4 для сложных вопросов). API Perplexity, по доступным сведениям, тоже не экспонирует параметры творчества: запросы настроены на оптимальный режим получения фактов. Это связано с позиционированием сервиса как «answer engine» с правдивыми ответами. Тем не менее, разработчик может частично управлять поведением запросов: например, задавать контекст, уточняющие вопросы. В целом Perplexity менее конфигурируем по стилю ответа, зато надежен в плане фактической информации.

Отслеживание запросов, ограничения: Бесплатная версия не ограничивает число запросов (но может быть суточный лимит на очень интенсивное использование, по опыту – сотни вопросов в день можно задавать без проблем). В платной подписке Perplexity Pro предоставляется до 300 запросов Pro-вопросов в сутки, приоритетный доступ и расширенный анализ файлов. Pro-пользователи также могут загружать неограниченно собственные файлы (PDF, DOCX) для анализа – модель их резюмирует и отвечает на вопросы по содержимому. Токеновые ограничения зависят от используемой LLM: при бесплатном использовании чаще работает модель типа Claude Instant или аналог (с контекстом ~100k токенов), а при Pro-запросах – GPT-4 или Claude 2 (контекст 8k–100k). Переполнение контекста маловероятно при типовых вопросах. Отслеживание запросов доступно через личный кабинет (для Pro-подписчиков есть статистика). В бесплатном режиме явного счётчика нет, но если сервис заметит слишком частые обращения, может временно замедлять ответы.

Интеграция и CI/CD: Perplexity ориентирован на интерактивное использование, однако API позволяет включить его в программные конвейеры. Например, можно написать autotest, который отправляет модель вопрос и проверяет, что среди источников есть ожидаемый URL. В контексте курса промпт-инжиниринга Perplexity ценен для демонстрации связывания LLM с поиском: студенты могут увидеть, как модель извлекает информацию в реальном времени. В CI-пайплайне Perplexity может выступать для проверки

фактов, хотя нужно учитывать расход запросов (Pro API платный). Особенность – высокая стоимость API: на Reddit сообщали, что Perplexity API считала токены результатов поиска как входные, из-за чего затраты были ~20x vs OpenAI по некоторым запросам. Поэтому в автоматизации стоит ограничивать объем веб-поиска или использовать ее выборочно.

Мультимодальность: Perplexity AI поддерживает текстовую и графическую выдачу. В Pro-режиме интегрирован генератор изображений на базе DALL-E 3 – пользователь может запросить иллюстрацию, и сервис создаст картинку. Однако ввод изображений или аудио не поддерживается: Perplexity не «понимает» картинки, а лишь генерирует их при запросе. Также сервис умеет анализировать загруженные текстовые файлы (как упоминалось, в Pro – безлимитно, в бесплатной версии – ограниченно). Иными словами, Perplexity – мультимодальный в части вывода (текст + сгенерированные изображения, цитаты, диаграммы), но не в части ввода. Уровень аудио- или видео-анализа отсутствует. Тем не менее, генерация изображений может быть полезна в обучении – например, сервис может наглядно показать график или схему по запросу студента.

Условия доступа (free/pay): Perplexity предоставляет полноценный бесплатный доступ к базовому функционалу: задавай вопросы, получай ответы с источниками. Подписка Pro ($20/мес) добавляет улучшенные модели (GPT-4, Claude 2), больше дневных запросов, анализ файлов и изображений. Для студентов этого курса базовой версии, как правило, достаточно – она уже покрывает большинство запросов. Если требуются сложные вопросы к GPT-4, можно рассмотреть приобретение Pro одной учетной записью для класса и демонстрировать через проектор. Однако оплата из РФ затруднена (нужна зарубежная карта). Академических лицензий специально для Perplexity пока нет. Отметим, что некоторые энтузиасты из РФ предлагают посреднические сервисы для оплаты Perplexity Pro, но в официальных рамках курса это вряд ли необходимо.

Grok (xAI)

Доступность в России: Grok – чатбот, разработанный компанией xAI (США) под руководством Илона Маска – официально не запущен на российском рынке. Изначально доступ к Grok давался только подписчикам X Premium (Twitter) в США с ноября 2023. К началу 2024 г. Grok был открыт для всех премиум-пользователей X, а позднее (август 2025) ограниченно стал доступен и бесплатно на платформе X. Однако сам X (Twitter) заблокирован в РФ, и даже при обходе потребуется учетная запись X Premium+ ($40 в мес) для полноценного доступа к последней версии Grok. Также есть отдельное мобильное приложение Grok (запущено в США в 2025), но российским пользователям оно недоступно через официальные сторы. Таким образом, прямой доступ к Grok в РФ сильно затруднен – необходимы VPN, иностранный аккаунт и оплата подписки (что выходит за рамки типичного курса). Тем не менее, первая версия Grok-1 была open-source (Apache 2.0), и её веса доступны на GitHub. Энтузиасты могут запустить Grok-1 локально, но эта модель слабее современных.

API и формат взаимодействия: xAI не предоставляет публичного API Grok всем разработчикам (на 2025 г. API возможно доступен партнерам). Grok интегрирован преимущественно в экосистему X (Twitter) и Tesla. Он работает как чат-бот: ввод – текстовые сообщения (в диалоге), вывод – ответ от лица ассистента. В интерфейсе X, Grok умеет выполнять веб-поиск и генерацию изображений по команде. Формат взаимодействия: как в ChatGPT, но с отличием – Grok имеет «юмористический, дерзкий тон», что задекларировано Маском. Если использовать open-source Grok-1, то через соответствующую библиотеку (подобно LLaMA) его можно встраивать и отправлять сообщения. Однако основная «официальная» платформа – внутри соцсети X и приложений.

В целом, без Premium-доступа мы не можем программно управлять Grok, что ограничивает его учебное применение.

Настройка генерации: Пользователю в интерфейсе X почти не предоставляются настройки (нет ползунков temperature и т.п.). Grok по умолчанию старается давать максимально развернутые и актуальные ответы. Известно, что Grok-2 получил режим «Think» для сложных задач – он активирует углубленное рассуждение. Возможно, в чате это переключается как особая команда. В версии Grok-3 были добавлены расширенный веб-поиск и возможности редактирования изображений. Но ручной контроль параметров (креативности, длины) конечному пользователю не предусмотрен. Если же запускать Grok-1 локально, то через используемую библиотеку можно задавать temperature, top_p, etc., как у любой модели.

Отслеживание запросов и лимиты: Поскольку Grok предоставляется как часть платной подписки, xAI вводила ограничения по числу запросов в зависимости от тарифа. Например, в раннем доступе бета-версии лимитировалось ~100 сообщений в сутки. Позднее, в июле 2025 года, Grok 4 открывали на короткое время для бесплатных юзеров, чтобы привлечь аудиторию. В интерфейсе X есть индикатор “GPT-4 equivalent” с описанием, но деталей лимитов публично мало. Можно предположить, что при пиковой нагрузке ответ от Grok может задерживаться. В data center xAI задействовано ~200 000 GPU для обучения Grok 3, так что вычислительных ресурсов много. Однако отсутствие свободного API и закрытость платформы не позволяют интегрировать Grok в собственные приложения и отслеживать токены. Сам X при общении с ботом может показывать сообщения о превышении лимита (как это бывает у ChatGPT).

Интеграция в CI/CD: На текущий момент интеграция Grok в автоматические процессы непредусмотрена (нет общедоступного API, SDK только для своего приложения). Теоретически можно использовать эмуляторы X API или парсить веб-версию X, чтобы получать ответы Grok, но это ненадежно и нарушает правила. Поэтому в курсе промпт-инжиниринга Grok больше выступает как объект для анализа, а не инструмент лабораторной работы. Можно на слайдах сравнить ответы Grok (взять примеры из обзоров) с другими моделями, но учащиеся напрямую с ним работать скорее всего не смогут. Исключение – использование открытой модели Grok-1 в локальном формате: ее можно попробовать запустить, например, в Google Colab (если есть 2×RTX A100, т.к. контекст 128k требует много памяти). В учебных целях это сложновато, поэтому в CI/CD pipeline Grok, как правило, не фигурирует.

Мультимодальность: Grok – мультимодальный ассистент. Уже версия Grok-2 (август 2024) умела генерировать изображения (через интеграцию с моделью Flux от Black Forest Labs). Grok-3 (март 2025) добавил функции редактирования изображений (AI-image editing). То есть пользователь может отправить картинку и задать команду отредактировать – бот возвращает измененное изображение. Также Grok подключен к реальному времени: он может делать веб-поиски, читать новости, анализировать тренды. На октябрь 2024 объявлялся Grok-1.5 Vision с возможностью понимать фото, скриншоты и т.п., но в публичный доступ та версия не вышла. Вероятно, эти наработки вошли в Grok-2 и выше. Кроме того, Grok интегрирован в Tesla – возможно, обрабатывает данные с камер (но этой информации мало). Аудио-входа/выхода пока не заявлено, хотя логично ожидать голосовой интерфейс (в самом X-app есть, возможно, опция озвучки). Таким образом, Grok представляет самую передовую мультимодальность среди конкурентов: текст, поиск, генерация картинок и их редактирование – всё в едином боте. Но без официального доступа нам сложно использовать эти возможности в учебных проектах.

Llama 2 (Meta AI)

Доступность в России: Llama 2 – серия открытых моделей от Meta (ранее Facebook) – доступна всем желающим без географических ограничений. Весы моделей Llama 2 (7B, 13B, 70B параметров) выложены для скачивания в июле 2023 г. под свободной лицензией. Лицензия разрешает коммерческое использование, за исключением продуктов с аудиторией более 700 млн пользователей (ограничение против гигантов-конкурентов). Для образовательных целей Llama 2 абсолютно бесплатна. Российские вузы могут самостоятельно развернуть Llama 2 на своих серверах или воспользоваться готовыми онлайн-репозиториями (Hugging Face, облачные GPU). Важное преимущество – никакого VPN или оплаты не требуется, всё открыто. Пользователи из РФ активно запускали Llama 2: известны случаи разворачивания модели даже на смартфонах. В курсе можно официально использовать Llama 2 без риска нарушить чьи-либо правила.

API и формат взаимодействия: Поскольку Llama 2 – это модель с открытым исходным кодом, единого облачного API нет (Meta предоставляет лишь ссылки на весы). Однако экосистема быстро подхватила модели: появились HuggingFace Inference API, репозитории с трансформерами (Transformers), поддержка Llama 2 в LangChain и других фреймворках. В итоге можно взаимодействовать с Llama 2 так же, как с ChatGPT, – передавая в модель последовательность сообщений или подсказку с промптом. Существует версия Llama 2-Chat – специализированно дообученная на ведение диалога, у неё формат ввода: <s>[INST] Вопрос пользователя [/INST] Ответ... (внутренний маркировочный стиль). Эти детали скрыты при работе через высокоуровневые библиотеки. Иными словами, интеграция Llama в приложении сводится к установке модели и вызову метода generate() с заданным текстом. Некоторые сервисы (например, Azure AI в 2024 г. добавил Llama 2) позволяют вызывать модель по API, но можно и локально. Для курса, вероятно, проще использовать Llama 2 через облачные ноды – например, запущенная на Google Colab или локальном сервере кафедры. Формат вывода – обычный текст. Ввод может включать несколько сообщений (сами можно объединить их в один промпт). Гибкость максимальная, никаких проприетарных ограничений.

Настройка генерации: При работе с Llama 2 доступен весь спектр параметров генерации. Можно задавать temperature, top_p, top_k, репетиционный штраф и т.д. Более того, можно дообучать (fine-tune) модель под свои задачи. Для преподавателя это означает, что Llama 2 можно «натаскать» на конкретные учебные данные (например, загрузить в нее конспекты лекций) и получать ответы, учитывающие именно этот материал. Такой подход продемонстрирует студентам технику дообучения LLM. Также Llama 2 поддерживает большие prompts: можно передавать в контекст сразу всю необходимую информацию (например, текст задачи, примеры решений). Нет внешних модераторов – разработчик сам решает, какую температуру или стиль задать. Например, для творческих задач можно увеличить randomness, для четких ответов – снизить. Отсутствие цензуры позволяет генерировать любой текст, но преподавателю следует соблюдать этику и следить, чтобы модель не использовалась для противоправных запросов. Педагогический плюс – студенты могут экспериментировать с параметрами и видеть, как те влияют на ответы.

Отслеживание запросов и ограничения: При локальном развертывании нет встроенных квот – всё ограничено лишь доступными вычислительными ресурсами. Llama 2-7B можно запускать на GPU с 12 GB VRAM (даже на игровой видеокарте), 13B – на 24 GB, 70B требует 2×48 GB или 8×A100 80GB в оптимальном случае. Впрочем, существуют сжатые версии (8-bit quantization) для запуска 70B на одном 48 GB GPU. Для образовательных целей масштаб 7B/13B вполне достаточен – они могут работать и на CPU (с библиотеки llama.cpp, пусть и медленно). Контекстный окна Llama 2 – 4096 токенов (примерно 3 тыс. слов). Существуют модификации с расширенным контекстом (например,

до 16k), но из коробки 4k. Лимитов на длину ответа как таковых нет – модель отвечает, пока не достигнет max_tokens или не завершит мысль. В отличие от API-сервисов, здесь нужно самому отслеживать выход за границы (например, обрезать слишком длинный ответ, если нужно). Для мониторинга запросов можно логировать обращения у себя, так как никакой веб-интерфейс не предоставляет статистику. Преподаватель может организовать для студентов общий доступ к Llama через интерфейс вроде Hugging Face Chat (где можно деплоить модель и давать ссылку) – там будет видна нагрузка и история.

Интеграция в CI/CD: Llama 2 очень хорошо подходит для CI/CD интеграции. Будучи локальной моделью, она позволяет писать юнит-тесты для промптов, которые всегда будут повторяемы, если зафиксирован сид генератора (в contraste с облачными, обновляемыми моделями). Например, можно сделать тест: «Дан промпт X – проверяем, что ответ содержит слово Y». С Llama это выполнимо – модель не поменяется внезапно завтра (если не заменить веса). Также Llama 2-Chat обучена следовать инструкциям, поэтому можно протестировать разные шаблоны промптов. В MLOps существуют инструменты типа DeepSpeed MII для ускоренного инференса Llama, что ускорит автотесты. Единственное, на что стоит обратить внимание – время генерации: на CPU 13B модель генерирует медленно (~1–2 токена/сек), лучше иметь GPU. Если возможности ограничены, можно использовать более мелкие аналоги (Mistral 7B, RWKV) для идеи, а Llama – для финального показа. В целом, свободная лицензия Llama 2 делает её идеальным кандидатом для песочницы в CI/CD: можно создавать любые сценарии и даже дописывать внутренности модели (но это уже для продвинутых).

Мультимодальность: Базовые версии Llama 2 – только текст. Однако экосистема предлагает дополнения: имеются проекты Llama 2 + Vision, например LLaVA (Large Language and Vision Assistant), где Llama 2 объединена с визуальным энкодером для понимания изображений. Meta отдельно выпускала модель LLaMA Adapter для обработки изображений. Но официально Llama 2 из коробки не умеет видеть картинки или слышать аудио. Что касается генерации изображений, Llama 2 может встраиваться в мультимодальные пайплайны: например, текстовая LLM генерирует описание, а диффузионная модель рисует. Meta анонсировала, что будущие версии (Llama 3?) будут нативно мультимодальными, пока же требуется доработка сторонними инструментами. В 2024 г. вышел прототип Llama 2 Text to Music, но широкого применения он не получил. Таким образом, для нашего курса Llama 2 выступает как мощная текстовая модель, на которой можно демонстрировать все принципы промпт-инжиниринга. Для работы с изображениями или звуком в заданиях надо дополнительно привлекать другие модели (CV или speech-to-text) – это выходит за рамки LLM. Можно, однако, показать студентам Llama 2 в связке с vision-моделью (пример: отправить картинку через BLIP-2 для описания, а Llama на основе описания ответит на вопрос – такой многошаговый workflow).

Условия бесплатного/платного доступа: Llama 2 – полностью бесплатна. Meta предоставила её для всех, кто согласится с лицензией. Для загрузки весов с официального сайта требовалась регистрация и согласие, но в открытых репозиториях модель есть без формальностей. Нет никаких лимитов на использование (кроме упомянутого ограничения на гигантские приложения с аудиторией >700M, что курсу не грозит). Нет платных планов, нет «академической» версии – она и так свободная. Более того, имеется множество fork-версий Llama 2 (например, от сторонних организаций), иногда даже более оптимизированных. При желании университет может развернуть Llama 2 на своем кластере и предоставить студентам через веб. В локальных заданиях студенты могут запускать небольшие версии на личных ноутбуках. Это существенный плюс: не требуется бюджет на API.

Qwen (Alibaba Cloud)

Доступность в России: Qwen – серия крупных моделей от Alibaba (Китай) – распространяется свободно как open-source. В августе 2023 Alibaba открыла веса Qwen-7B и Qwen-14B (а также диалоговые версии Qwen-Chat) для всех желающих. В декабре 2023 добавлены Qwen-1.8B и даже Qwen-72B, а также аудио-модели. Они доступны на GitHub и HuggingFace под благоприятной лицензией (Apache 2.0). Таким образом, изначальных барьеров для РФ нет – модели можно скачать или запустить в облаке. Более того, Alibaba Cloud (АлиЮнь) предлагает сервис ModelScope, где Qwen доступна для использования через API. Российскому разработчику ничто не мешает зарегистрироваться на Alibaba Cloud (санкции Китая не вводил) и использовать Qwen через их сервисы. Однако интерфейс АлиЮнь на английском/китайском, придется разобраться. Для упрощения можно использовать Qwen, развернутый локально или через huggingface.co (Inference API там доступен всем, но с ограничениями скорости). В целом, доступность Qwen для нас сопоставима с Llama 2 – полный open-source, плюс опция облачного API от Alibaba.

API и формат взаимодействия: Qwen спроектированы с совместимостью OpenAI API. Например, в документации Alibaba Cloud указано, что Qwen-Chat поддерживает тот же формат ролей и сообщений. Более того, Qwen интегрирован в OpenAI-совместимые платформы (OpenRouter etc.). Если запускать модель локально, можно использовать HuggingFace Transformers: там есть модель Qwen-14B-Chat с примером, как подавать промпт. Формат ввода: китайско-английский. Модель тренирована на двуязычных данных, так что русский язык для нее чужой – он понимается, но хуже. Оптимально использовать Qwen для заданий на английском или китайском. Что интересно, Alibaba открыла Qwen API в своей среде: разработчики могут вызывать Qwen через REST API Alibaba (видимо, платно после free trial). Но в рамках курса, вероятно, удобнее локально (если есть GPU) или через HF spaces. Формат взаимодействия – стандартный: текст на вход, текст на выход, можно вести последовательность (но нужно самому склеивать, open-source модель контекст держит внутри только что подано).

Настройка генерации: Как open-model, Qwen допускает полную настройку параметров. Temperature, top_p, max_length – все регулируется при генерации через код. Примечательно, что Alibaba акцентирует возможности fine-tuning: выпускалась версия Qwen-Chat с 1M дополнительных инструкций обучений, есть версия Qwen 2.5 с улучшениями. Для нас это означает, что можно при желании дообучить Qwen на русском корпусе, если потребуется. Но вряд ли в рамках курса мы пойдем на такой трудоемкий процесс. Скорее, мы можем продемонстрировать настройку параметров: запустить Qwen-7B с разными температурами и показать разницу.

Также Qwen известны длинным контекстом: Qwen-14B поддерживает 8192 токенов, а Qwen-14B-Inf (с суффиксом) – до 16 384 токенов. В январе 2024 Alibaba выпустила Qwen-2.5-Max (размер не указан, но судя по названию – большая модель), которая заняла 7-е место в рейтинге Chatbot Arena, сравнявшись с топ проприетарными LLM. Новейшая Qwen 2.5-Omni-7B (март 2025) имеет уже мультимодальную архитектуру, но о ней ниже. Настройка генерации Omni-версии включает выбор модальностей (например, можно отключить анализ видео, чтобы быстрее ответить).

Отслеживание запросов и лимиты: Если Qwen используется локально – лимитов нет, только ограничения железа. Qwen-14B требует 28 GB видеопамяти в fp16, но есть 4-bit quant версия (~8 GB). Qwen-7B и вовсе легко запускается на 8 GB GPU. HuggingFace Inference API для Qwen-14B-Chat существует, но он может быть медленным и с ограничением запросов в минуту (для бесплатных). Alibaba Cloud ModelStudio предлагает Qwen через API с тарификацией (по токенам или часам использования, можно найти цены на

alibabacloud.com). Вероятно, вход 1k токенов $0.001, выход 1k $0.005 (Alibaba стремится демпинговать OpenAI).

В общем, для учебных целей проще считать, что Qwen бесплатен (если запускать самим), но придется контролировать вычислительную нагрузку (чтобы ноутбуки не зависли). Если все студенты одновременно начнут слать большие тексты на Qwen-14B на одной машине – будут тормоза. Надо или раздать каждому маленькую модель (7B) или выделить кластер.

В плане отслеживания – open-source модель не ведет логов, так что преподавателю стоит самому заложить логирование (кто какой запрос отправил, какой ответ получил).

Интеграция в CI/CD: Qwen очень перспективен для интеграции. Компания Alibaba выложила оптимизированный фреймворк для инференса (с поддержкой ускоренного вывода, и даже Mixture of Experts в Qwen-2). В 2025 выпущена Qwen2-Audio и Qwen2-Omni, и Alibaba говорит, что на их базе удобно строить “агентов” для бизнеса. Для нашего CI Qwen ничем не хуже Llama: он локальный, реплицируемый, управляемый. Его плюс – многоязычность (китайский/английский), но это же и минус – для русского out-of-the-box слабее. Зато Qwen-Chat умеет выдавать формализованные ответы (скажем, списками) и немного лучше структурирован (подмечено в обзорах).

CI/CD pipeline с Qwen: аналогично Llama, можно фиксировать seed. С учетом того, что Alibaba сама тестирует Qwen на куче задач, она достаточно детерминирована (при temp=0) – отлично для юнит-тестов промптов.

Мультимодальность: Одно из главных достоинств Qwen – полная мультимодальность в серии 2.5. Alibaba в сентябре 2024 запустила Qwen-2.5 (усовершенствование архитектуры), а к марту 2025 представила Qwen-2.5-Omni-7B – унифицированную модель, понимающую текст, изображения, аудио и видео. Это фактически аналог задуманного Google Gemini: единая нейросеть для всех типов данных. Qwen-Omni может принимать на вход текст вместе с картинкой или аудио, и на выходе генерировать текстовый ответ или даже голос (причем, как указано, натуральный голос в реальном времени). То есть Omni встроила синтез речи – уникально. В описании говорится, что модель способна “process diverse inputs, including text, images, audio, and videos, while generating real-time text and natural speech responses”. Это ставит новый стандарт: Qwen-Omni-7B – сравнительно небольшая (7 млрд), но обучена на мультимодальных парах (изображение+описание, видео+звук+описание и т.д.). Она открыта на HuggingFace и GitHub.

Конечно, для локального запуска Omni нужны одновременно GPU и аудио/видео данные – не тривиально. Но можно показать студентам демо (Alibaba наверняка выложила примеры: навигация для незрячих с описанием окружающего, или рецепт по видео продуктов).

Помимо Omni, есть Qwen-Audio – модель для аудиовопросов (Speech LLM), и Qwen-VL – визуально-лингвистическая модель (аналог GPT-4V). Всё это открыто. Qwen-VL-Chat может, например, отвечать на вопросы по картинке (с выдающейся точностью).

Для курса эти возможности значат: мы имеем доступ к передовой мультимодальности безплатно. Если инфраструктура позволяет, можно дать практическое задание – например, запустить Qwen-VL на сервере, и пусть студенты отправляют ей картинки и получают ответы. Это продемонстрирует интеграцию зрения и языка.

Также можно показать Qwen-Audio: дать фрагмент музыки – модель определит инструмент или жанр (так заявлено).

Конечно, всё это сложнее, чем с ChatGPT, но если есть мотивация, Qwen предоставляет необходимые модели.

Условия доступа: Вся серия Qwen 1.x и 2.x – open-source (Apache 2.0). Alibaba даже хвастается, что открыла более 200 моделей за последние годы. Коммерческое использование разрешено. Единственное: Alibaba предлагает коммерческие улучшенные версии через свой облако, которые отличаются от открытых более свежими фичами. Но мы можем оперировать открытыми без ограничений. Нет платных планов (если не брать Alibaba Cloud услуги).

Для курса: просто скачиваем нужную версию Qwen, никаких регистраций. Если хочется Alibaba Cloud API – нужно иметь аккаунт и карту (Китай не запрещает россиянам, так что возможно), но процесс может быть сложнее, чем играться с open model.

Claude (Anthropic)

Доступность в России: Claude 2 – флагманская модель компании Anthropic (США) – официально доступна только в некоторых странах (США, Великобритания, и др.). России нет в списке разрешенных – при попытке зайти на claude.ai появляется сообщение о недоступности региона. Таким образом, без VPN не обойтись. Кроме того, для регистрации на сайте требуется номер телефона из поддерживаемых стран и почта Gmail. Российские номера не принимаются. Тем не менее, существует несколько обходных путей, например использование ботов-посредников, которые отправляют запросы в Claude от вашего имени (GPTunnel и др.). Но в рамках курса такие методы неофициальны. API Anthropics также ограничен: доступ предоставляется через ожидание в waitlist и требует иностранной оплаты. На конец 2024 г. некоторые российские энтузиасты получили доступ к Claude API используя виртуальные номера и VPN, но это довольно сложно. Вывод: прямого легального доступа у студентов из РФ нет. Возможно, стоит ограничиться теоретическим рассмотрением Claude или, если очень нужно практическое – воспользоваться готовыми датасетами с ответами Claude.

API и формат: Claude предоставляет API, аналогичный OpenAI Chat API (модель claude-2 или claude-instant). Формат – список сообщений с ролями system, user, assistant. Интерес Anthropic – модель обучена следовать письменным инструкциям (“constitution” approach), поэтому она даже без system prompt ведет себя этично. Через API можно отправлять также файлы – особенность Claude 2 в том, что он принимает вложения (например, PDF) и может на них отвечать. Это реализовано как часть API (веб-интерфейс позволяет прикрепить файл). В использовании формат удобный: просто текст. Если бы у нас был доступ, мы могли бы интегрировать Claude так же, как ChatGPT. Но, учитывая сложности, скорее всего, мы не будем напрямую с ним работать, а только расскажем студентам.

Настройка генерации: Anthropic API позволяет задавать параметры: max_tokens_to_sample, temperature, top_p, stop_sequences и др. То есть гибкость примерно как у OpenAI. Отличие – “constitution”: вместо классического RLHF у Claude набор правил, которые он старается соблюсти (не быть токсичным, не выдавать личные данные и пр.). Это делает его более предсказуемым в плане безопасности. Например, Claude известен тем, что охотнее дает подробные пошаговые решения задач, меньше отказывается говорить на сложные темы (в сравнении с ChatGPT).

Конкретно, temperature можно регулировать (от 0 до 1), но даже на высоких значениях Claude обычно сохраняет связность.

Gemini (Google DeepMind)

Доступность в России: Gemini – семейство моделей нового поколения от Google – не доступно российским пользователям напрямую. После слияния Google Brain и DeepMind проект Gemini анонсировали как ответ GPT-4. Запуск состоялся в декабре 2023: представлены модели Gemini Ultra, Pro, Nano. Однако Bard (Google AI чат), в который

интегрирован Gemini Pro, официально не работает в РФ. Даже в Европе запуск задерживался из-за GDPR, а Россию Google, покинув рынок, не включает в поддерживаемые локации. Таким образом, для доступа нужен VPN, аккаунт Google (который, впрочем, у всех есть), но и этого может быть мало – Google может по геоданным аккаунта блокировать Bard. Впрочем, некоторые пользователи сообщали, что через VPN удавалось пользоваться Bard. Но если Google ужес точит, то никак. Через Google Cloud Vertex AI также: Россия под санкциями, предоставление сервисов ИИ запрещено. Поэтому прямого доступа к Gemini у нас нет. Тем не менее, некоторые компоненты Gemini просочились в открытый доступ: например, в феврале 2024 Google выпустил маленькие open-source модели Gemma (2B и 7B) – их назвали «облегченной версией Gemini». Они доступны миру, но они намного слабее топовых Gemini Ultra. В курсе, скорее всего, Gemini будет фигурировать только описательно.

Manus

Доступность в России: Manus AI – сравнительно новый китайский проект, прославившийся вирусным ростом в марте 2025. Это AI-ассистент (целостный агент), который позиционируется для молодежи. Особенность Manus – полностью бесплатный и беспрепятственный доступ: не требуется ни VPN, ни регистрация. Пользователь просто заходит на сайт или в приложение Manus и начинает общаться. Поскольку разработчик – китайский стартап, и они нацелены на массовость, они не стали вводить барьеров. В России Manus пока не очень известен, но, по данным китайских источников, он работает через глобальное веб-приложение (есть англоязычная версия). Так что технически студент из РФ может открыть Manus.im и попробовать. (Если вдруг сайт заблокирован РКН – маловероятно, он не политический ‑ тогда через VPN). В мобильных магазинах, возможно, приложение недоступно в RU-регионах, но веб решает.

API и формат взаимодействия: Manus AI – прежде всего чат-бот с веб/мобильным интерфейсом. На март 2025 у него нет публичного API, хотя разработчики планируют его добавить для монетизации (см. ниже). Поэтому интегрировать Manus в свои программы пока нельзя. Формат общения – диалог, очень схожий с ChatGPT. Поддерживаются английский и китайский, на этих языках Manus работает “плавно, с естественным дружелюбным тоном”. Русский язык Manus воспринимает хуже – он не заявлен как целевой, но может что-то перевести. Важная особенность: Manus позиционируется не только как Q&A бот, но и как “универсальный помощник, выполняющий задачи”. То есть у него есть функции-агенты: например, помочь написать эссе, решить задачу, сгенерировать идею.

Сравнение по качеству и эффективности

Рассмотрим платформы в разрезе четырех важных критериев: релевантность ответов, воспроизводимость результатов, конфигурируемость и скорость генерации.

Релевантность (точность и соответствие ответов запросу): большинство ведущих LLM (ChatGPT, DeepSeek, Claude, Gemini) обеспечивают высокую релевантность, особенно на английском языке. GPT-5 от ChatGPT часто признается эталоном точности, однако на русском языке Claude иногда дает более естественные формулировки, а специализированные модели (GigaChat, YandexGPT) лучше справляются с локальными реалиями. Например, GigaChat Max на независимом русскоязычном лидерборде MERA – в числе лидеров, уступая лишь GPT-4 условной открытой версии. YandexGPT оптимизирован под русские запросы и, интегрируясь с Поиском, снабжает ответы актуальными данными,

повышая их объективность. DeepSeek, по отзывам, выдает содержательные и академически точные ответы (его сравнивают с ChatGPT на равных). Manus AI в силу ориентации на простые запросы студентов может давать менее глубокие, хотя и более развлекательные ответы – т.е. релевантность чуть жертвуется ради дружелюбия. В целом, для сложных профессиональных вопросов (программирование, наука) наиболее релевантны Claude 2 и GPT-5, затем DeepSeek и GigaChat Max. Для фактических вопросов с актуальными данными – Perplexity и YandexGPT (так как они ищут в интернете). Стоит отметить, что Gemini Ultra в тестах превзошел GPT-4 на многих бенчмарках, но это скорее перспективы – мы не можем напрямую подтвердить релевантность Gemini в учебных задачах из-за недоступности.

Воспроизводимость (повторяемость результатов): под этим критерием подразумевается, насколько одинаковые ответы получаются при повторном запуске и насколько поведение модели стабильно с течением времени. Здесь преимущества у open-source моделей: Llama 2, Qwen, open Grok – они не меняются, если вы используете фиксированные веса, и можно задать фиксированный random seed, получив детерминированный ответ. Это идеально для воспроизводимых экспериментов: например, чтобы сравнить варианты промпта, исключив случайность. Коммерческие API (OpenAI, Anthropic) не позволяют явно задавать seed, и ответы могут меняться от запроса к запросу (при ненулевой temperature). Кроме того, они периодически обновляют модели: так, ответы ChatGPT GPT-3.5 летом 2023 и зимой 2024 могли различаться из-за скрытого обновления модели. Claude 2 и Perplexity могут менять контент-стратегию (например, Anthropic повысит строгость фильтров – некоторые ранее доступные ответы станут недоступны). С этой точки зрения, самая воспроизводимая – Llama 2 (локально, с seed=0 всегда один ответ), затем Qwen (то же условие). GigaChat и YandexGPT меняются реже – у них крупные релизы (GigaChat 1.0 → 2.0), но внутри версии ответы достаточно стабильны. У GigaChat API при temperature=0 можно получить почти детерминированный стиль (но все же мелкие варьирования пунктуации возможны). ChatGPT GPT-4 API при temperature=0 обычно тоже дает повторяемый ответ – OpenAI заявляет, что temperature=0 делает вывод детерминированным, хотя на очень длинных текстах возможны незначительные отличия. Manus AI, напротив, не гарантирует повторяемость: он настроен на креативность, и два запуска могут дать разные шутки или формулировки. В итоге, для исследовательских работ, требующих строгой повторяемости, лучше использовать локальные модели (Llama, Qwen) или ChatGPT/Claude с temp=0 и фиксированными версиями API. Для студентов, когда важно, чтобы пример из методички генерировал тот же результат у всех – опять же, либо локальная модель, либо предоставить снапшот ChatGPT (OpenAI позволяет указать версию модели по дате).

Конфигурируемость (возможность настройки и кастомизации): под этим подразумевается как низкоуровневое управление параметрами генерации, так и высокоуровневое – дообучение, подключение инструментов. Здесь лидеры – open-source: Llama 2 можно дообучить на своих данных, изменить архитектуру, добавить модули для изображения – полная свобода. Qwen также открыта для тонкой настройки, и Alibaba даже снабдила её примерами дообучения под диалоги, под аудиовход, под длинный контекст. Grok-1 (опен) тоже можно переподготовить, хотя он уже устарел. Среди закрытых платформ наиболее гибкая – OpenAI: их API дает много параметров, плюс поддерживает функции (function calling), а с 2023 г. – и fine-tuning определенных моделей (GPT-3.5 можно дообучить под конкретный стиль). Anthropic пока не предлагает паблик-фаинтюна, но Claude можно “кастомизировать” большой системной инструкцией (Constitution вшита, но вы можете добавить свои правила в System, и он их учтет). Google Vertex (Gemini) вероятно

позволит fine-tune модели под задачи – PaLM 2 уже такое имелось (Adapter tuning). Российские – GigaChat и YandexGPT – частично конфигурируемы: у Сбера нет открытых весов, но они дают инструменты для обучения своих ботов на базе GigaChat (через SberCloud, используя технику RAG или привязки данных). Яндекс Cloud объявил AI Assistant API – это, по сути, платформа, где можно настроить своего чат-бота, обучив YandexGPT на доменных данных (например, на знаниях компании). То есть fine-tune как услуга. Поэтому преподаватель может обратиться к отечественным платформам для кастомных кейсов – возможно, под образовательные нужды Яндекс или Сбер выделят доступ к такой настройке. ChatGPT частично конфигурируется "Системным сообщением" – это поверхностно, но всё же влияет. Manus AI наименее гибок: у него фиксированный "модный" тон, и возможностей подстроить под серьёзный стиль нет (это его фишка и ограничение). Perplexity тоже практически не настроить – он специально "честный ответ с ссылками", без вариативности. Вывод: open LLM ≫ API крупных LLM ≫ закрытые ассистенты по степени возможности конфигурации. Для курса, где может понадобиться настроить модель под задачи студентов (например, говорить всегда пошагово или отвечать только цитатами) – лучше всего подходят OpenAI/Anthropic API (можно задавать параметры и примеры) или собственная Llama.

Скорость ответа: быстродействие зависит от мощности инфраструктуры и размеров моделей. Здесь облачные сервисы обычно выигрывают, но есть нюансы. Самые быстрые отклики – у моделей, оптимизированных для реального времени: ChatGPT-3.5 Turbo отвечает практически мгновенно на короткие запросы, YandexGPT быстрый режим выдает короткие ответы за доли секунды (важно для поисковых ответов). GigaChat Lite специально сделан для скорости – жертвуя глубиной рассуждения. Manus AI тоже отметили за очень шустрый интерфейс без задержек (возможно, он использует относительно небольшую модель, чтобы держать скорость). Claude 2 на коротких запросах также достаточно быстр, но при длинном ответе (особенно с 100k контекстом) он может генерировать сравнительно медленно (но всё еще быстрее GPT-4). GPT-4 славится некоторой медлительностью: он генерирует ответ заметно медленнее GPT-3.5, особенно в чат-интерфейсе (OpenAI намеренно сдерживает скорость по соображениям нагрузки). В DeepSeek сравнение: он "отвечает чуть дольше" ChatGPT, но при этом "чуть умнее" – то есть небольшая задержка есть. Вероятно, DeepSeek-Chat (V3) ближе по скорости к GPT-4 (несколько секунд на сложный ответ). Perplexity может иметь задержку из-за веб-поиска: он сначала ищет (~1–2 сек), потом генерирует ответ (ещё пару секунд), так что суммарно иногда медленнее просто LLM. Grok – по отзывам, версия Grok-4 отвечала довольно быстро. Gemini Nano (на Pixel) – ультра-быстрый, но он ограничен по размеру модели (для простых запросов). Gemini Ultra, наоборот, вероятно медленнее GPT-4, ведь он ещё больше – но Google может параллелить выдачу за счёт MoE. Summing up: для коротких вопросов самым быстрым будет ChatGPT-3.5 или любой "Instant" (Claude Instant, GigaChat Lite) – ответы приходят за <1 секунды. Для больших текстов быстрее справляется Claude – он может выдать длинное продолжение очень эффективно, опережая GPT-4 (который часто ограничен по токен/минуту). Llama 2 локальная скорость зависит от железа – на хорошем GPU 70B может выдавать ~10 токенов/сек, что приемлемо, но GPT-3.5 на сервере может ~30 tok/sec.

In the classroom context, скорость влияет на интерактивность: на презентации лучше показать модели с быстрым откликом (ChatGPT-3.5, GigaChat), чтобы не ждать. Для длительных задач (на фоне, с анализом больших данных) – Claude с его 100k контекстом: хоть и не молниеносно, но он один из немногих, кто вообще справится без разбивки данных.

Рекомендации по выбору платформ для заданий

Исходя из проведенного анализа, можно дать следующие рекомендации по использованию LLM-платформ в учебном процессе курса «Промпт-инжиниринг в проф. деятельности»:

Для базовых заданий (освоение промпт-техник, индивидуальные эксперименты): рекомендуется использовать доступные и русскоязычные модели, чтобы студенты не тратили время на обход блокировок. Оптимальный выбор – GigaChat (веб-версия или бот), а также YandexGPT (через Алису или ya.ru чат). Эти модели свободно понимают вопросы на русском, дают достаточно качественные ответы и просты в использовании. GigaChat подойдет для творческих задач (написать рассказ, составить письмо) и для технических вопросов на русском. YandexGPT хорошо применять для заданий, требующих актуальной информации или интеграции с поиском (например: «сформулируй ответ на вопрос по недавно вышедшей статье» – Яндекс найдет и ответит). ChatGPT (GPT-3.5) также может быть использован (через VPN) для базовых упражнений, особенно если нужна английская практика, но его блокировка усложняет доступ – имеет смысл либо подготовить общий аккаунт для класса, либо ограничиться демонстрацией преподавателем. Manus AI можно привлекать как дополнительный инструмент: например, дать студентам сравнить ответы Manus и GigaChat на простые вопросы, чтобы они потренировались анализировать различия тональности и полноты. Однако Manus не должен заменить основные платформы, так как его качество на русском ниже и нет гарантий устойчивости работы (периодически может требовать кредиты).

Для сравнительных заданий (сравнение ответов разных моделей, оценка качества промптов): необходимо задействовать несколько платформ разных типов. Рекомендуется включить хотя бы одну открыто-закрытую пару западных моделей и одну пару российских/китайских:

ChatGPT (GPT-4) vs Claude 2: если есть возможность (у преподавателя через Poe или VPN), стоит показать разницу между этими двумя передовыми англоязычными моделями. Например, дать им сложный вопрос по анализу текста – студенты увидят, что GPT-4 ответит более структурированно, а Claude добавит “человечности” и может учесть больше контекста (Claude держит 100k контекста, GPT-4 только 8k). Такое сравнение можно провести демонстрационно (преподаватель показывает заранее полученные ответы, поскольку студенты сами не достучатся до Claude).

DeepSeek vs ChatGPT: оба позиционируются как универсальные помощники, но DeepSeek может чуть лучше знать китайский/азиатский контекст и выдает ссылки на китайские источники. Задание: сравнить, как ChatGPT и DeepSeek отвечают на вопрос, требующий знания, скажем, китайской истории (или просто фактологический вопрос). Ожидаемо, DeepSeek даст не хуже ответ, а возможно и более подробный – это покажет студентам, что не все ограничивается OpenAI, есть сильные конкуренты. DeepSeek также можно сравнить с GigaChat: в Habr отмечалось, что ChatGPT и DeepSeek лучше GigaChat по качеству ответов. Пусть студенты убедятся: например, вопрос по физике или математике – сравнить ответы GigaChat Pro и DeepSeek-R1. Вероятно, DeepSeek выстроит более чёткую логику решения, а GigaChat может чуть упростить. Это подведет к дискуссии о необходимости локальных доработок.

GigaChat vs YandexGPT: предложить чисто русскоязычные модели на один вопрос. Например, задать обоим: “Объясни парадокс кота Шрёдингера простыми словами”. Ожидается, что оба справятся, но по стилю будет различие: GigaChat (особенно Max) даст более строгий, научно выдержанный ответ, а Яндекс может написать проще, с житейским

примером (исходя из практики Alice). Обсудить, какой подход лучше для целевой аудитории.

Llama 2 vs ChatGPT-3.5: если есть техническая возможность, можно студентам дать поработать с локальной Llama 2 (например, через интерфейс HuggingFace или Colab). Задание: придумать промпт и получить ответ от Llama 2-13B и от ChatGPT. Сравнить, где ответы точнее. Скорее всего, на сложных вопросах Llama 2 может ошибаться или давать более краткий ответ – студенты увидят разницу между моделью с 13 млрд параметров и 175 млрд (GPT-3.5). Это подчеркнет важность масштаба и RLHF.

Manus vs Perplexity: любопытное сравнение – оба ориентированы на широкую аудиторию, но Manus отвечает без источников, а Perplexity – строго с фактами. Можно задать открытый вопрос (например, "Почему небо голубое?"). Manus, вероятно, выдаст популярно-научное объяснение с шутливым тоном, а Perplexity – короткий фактический ответ с ссылками на учебник. Обсудить, где легче воспринимать – "сухой" правильный ответ или дружелюбное пояснение. Так студенты поймут, как разные задачи (развлекательный бот vs справочная система) диктуют стиль.

Эти сравнительные упражнения помогут выработать критическое отношение к ответам ИИ и навыки оценки качества: релевантности, полноты, стиля. Главное – обеспечить, чтобы студенты имели хотя бы ограниченный доступ к упомянутым парам (кроме Claude/GPT-4, тут можно обойтись разбором готовых примеров).

Для мультимодальных заданий (работа с изображениями, аудио, сложными данными): лучше использовать платформы, специально поддерживающие такие возможности. Оптимальный выбор: ChatGPT Plus (GPT-4 Vision) и GigaChat Vision.

Задания с изображениями: например, дать студентам картинку с диаграммой и попросить модели ее проанализировать. ChatGPT Plus способен "посмотреть" изображение (в чате можно загрузить) и написать выводы. GigaChat Pro позволяет прикрепить иллюстрацию через API или веб – он определит, что на ней, и даже распознает текст на картинке. Студенты могут сравнить: GPT-4Vision опишет изображение на английском довольно подробно; GigaChat – на русском и, возможно, уделит внимание специфике (как в примере со стилем одежды). Если нужно только описать картинку, можно и YandexGPT подключить – правда, прямого функционала нет, но можно воспользоваться их CV API (это усложнит). Более легкий путь – Perplexity с DALL-E3: можно попросить Perplexity сгенерировать изображение по запросу или проанализировать (нет, анализ не умеет). Все же, ChatGPT и GigaChat – основные инструменты.

Задания с аудио: здесь выбор меньше. ChatGPT (в приложении) умеет воспринимать голос – можно на занятии голосом задать вопрос и показать студентам ответ. Но лучше сконцентрироваться на тексте. Если очень нужно – можно показать демо Qwen-Audio: например, проиграть кусочек музыки и рассказать, что модель (по документации) может распознать. Но вживую это сложно организовать. Проще – Яндекс: его переводчик или браузер уже умеет суммировать видео. Можно взять небольшое видео лекции (2-3 минуты) и показать, как Яндекс браузер (с включенным ИИ) выдает краткое резюме на русском. Это продемонстрирует практическую мультимодальность (а за ней стоит YandexGPT).

Комбинированные задания: например, "Загрузите PDF-файл статьи и получите от модели ее краткий обзор + сгенерируйте иллюстрацию к ней". Это можно сделать связкой Perplexity (анализ PDF) и ChatGPT (генерация рисунка) или DeepSeek (анализ файлов и

генерация изображений). Однако, из-за ограничений доступа, возможно лучше не давать слишком сложных заданий студентам, а показать в лекционном формате.

Рекомендуется хотя бы показать на проекторе работу GPT-4 Vision – как он отвечает на нарисованную задачу из математики или объясняет фотографию. Также и GigaChat Vision: можно открыть их веб-интерфейс, загрузить картинку (например, страницу текста, как в пресс-релизе) и получить вывод. Такие демонстрации вдохновляют и показывают актуальный уровень AI. Если инфраструктура в классе позволяет, некоторые мультимодальные активности можно сделать интерактивно (студенты предлагают картинку – модель отвечает). Только нужно заранее обеспечить необходимые доступы (ChatGPT Plus у преподавателя, GigaChat Vision доступ через sber.ru).

При выполнении лабораторных работ:

Для экспериментов с параметрами генерации (temperature, длина и т.п.) нагляднее использовать открытые модели (Llama 2, либо доступные через API). Можно предложить студентам небольшой скрипт в Google Colab, где вызов Llama 2 делается с разными настройками, и они наблюдают разницу в стиле. Это лучше, чем пытаться “на глаз” в ChatGPT уловить изменения, ведь ChatGPT UI не дает прямого контроля.

Если лабораторная подразумевает создание своего чат-бота, то в качестве бэкенда для него удобно взять либо OpenAI API (если вуз готов оплатить немного токенов и настроен VPN/proxy), либо Sber GigaChat API (для русскоязычного бота, и нет вопросов с оплатой, надо только зарегистрировать разработчика). YandexGPT API тоже возможен, но процедура доступа может быть сложнее. OpenAI API хорошо документирован и студенты быстрее разберутся. GigaChat API – преимущество в локальном языке и отсутствии санкц.преград. В итоге, в курсовых мини-проектах, где важна интеграция, можно ориентироваться на эти варианты.

Учебные примеры и корпуса: при планировании примеров стоит включать данные разных языков и областей, чтобы подчеркнуть сильные и слабые стороны моделей. Например, небольшой кейс на китайском языке: спросить DeepSeek или ChatGPT о китайской поговорке – DeepSeek может ответить лучше. Или код-пример: попросить ChatGPT и GigaChat написать функцию на Python – сравнить (обычно ChatGPT пишет более оптимизировано). Этический кейс: попросить разные модели сгенерировать потенциально нежелательный ответ (вежливо сформулировать провокацию) – увидеть, как ChatGPT откажет, Claude откажет более эмпатично, а, скажем, Manus или Llama могут попытаться ответить. Это демонстрирует встроенные правила и ответственность разработчиков.

Настройка автотестов и визуализация результатов

Автоматическое тестирование студенческих работ – ключевой элемент инфраструктуры курса. Корректная настройка автотестов и удобное представление их результатов позволяют студентам получать быструю обратную связь, а преподавателю – оперативно контролировать успеваемость. Ниже приведены рекомендации по настройке автотестирования и организации вывода результатов:

Разработка автотестов. Преподаватель (или разработчики курса) разрабатывает набор тестовых сценариев для каждой лабораторной работы или задания. Эти тесты могут проверять разные аспекты: правильность ответов нейросети на заданные промпты, корректность работы написанного студентом кода, соответствие формату вывода и др. При создании автотестов следует стремиться к максимальной автоматизации оценки – чтобы

минимизировать ручную проверку. Например, если задание – написать набор промптов под определенную задачу, автотест может автоматически отправить эти промпты в API выбранной LLM и проанализировать характер ответа (поиск ключевых слов, форматов). Если задание – написать скрипт, то автотесты запускают этот скрипт на тестовых данных. Важно: автотесты должны быть устойчивы к вариативности ответов нейросетей. То есть, вместо точного сравнения текста ответа лучше проверять на содержание некоторых ключевых фраз, длину ответа, отсутствие недопустимых элементов и т.д., либо задавать фиксированный seed/температуру=0 для детерминизма.

Интеграция с CI. В файле `.gitlab-ci.yml` настраивается job (например, `run_tests`), который запускает автотесты. Для Python-проекта это может быть: установка зависимостей (`pip install -r requirements.txt`), затем запуск `pytest --maxfail=1 --disable-warnings -q` (в тихом режиме) или другого тест-раннера. По завершении тестов необходимо сохранить результаты. Рекомендуется выводить отчеты в формате JUnit XML – многие тестовые фреймворки умеют генерировать такой файл (например, `pytest` с флагом `--junitxml=report.xml`). GitLab CI настроен так, чтобы этот файл считался артефактом job и помечался как тестовый отчет. Это позволит просматривать сводку тестов прямо в интерфейсе GitLab (число упавших/пройденных тест-кейсов).

HTML-отчеты. Для более наглядной визуализации результатов можно дополнительно формировать HTML-отчет. Например, использовать библиотеку pytest-html или Allure Report для генерации красивого отчета с детализацией по каждому тесту (включая логи, скриншоты и пр., если нужно). Процесс может быть таким: после прогона тестов (stage test) запускать отдельный stage (например, report), который собирает результаты. Если тесты выполнялись параллельно в нескольких job, можно склеить отчеты из каждого (объединить несколько XML в один или воспользоваться функциями Allure для агрегирования). Полученный HTML-файл (или папку со статическими файлами отчета) добавляется как артефакт сборки. В настройках GitLab CI можно указать, чтобы артефакты stage report не удалялись сразу (или публиковались через GitLab Pages). Например, некоторые компании размещают итоговые отчеты на GitLab Pages для удобного доступа, однако в учебных целях достаточно хранения артефакта внутри GitLab. Студент сможет скачать HTML-отчет из страницы pipeline (раздел Job Artifacts), открыть локально в браузере и просмотреть подробности: какие тесты пройдены, какие упали, какие ошибки возникли.

Для простых случаев можно обойтись и без HTML: GitLab отображает базовую статистику тестов и лог консоли, в котором видны сообщения `assertion failed` и т.п. Тем не менее, методически рекомендуется предоставить хотя бы краткий структурированный отчет, чтобы студент понимал, в чем ошибка. Например, можно в вывод теста явно писать: "Ошибка: функция X должна возвращать число, а вернула строку" и т.п. – тогда этот текст будет виден в логах pipeline.

Настройка уведомлений. Преподаватель может настроить оповещения о результатах CI: например, интегрировать GitLab с почтой или мессенджером (Rocket.Chat/Slack) для отправки сводки о каждом прогоне тестов. Однако в условиях учебного процесса это опционально. Достаточно, что студент сам видит статус своего конвейера. При необходимости, на очных занятиях преподаватель может просмотреть список последних pipeline в проектах студентов и выявить, у кого тесты красные (не прошли) – чтобы оперативно помочь.

Пример реализации: в шаблонном репозитории настроен параллельный запуск двух групп тестов (чтобы ускорить проверку). После прохождения, в stage report собирается единый HTML. Итоговый `.gitlab-ci.yml` включает артефакты (JUnit, HTML). Студент, зайдя в свой проект в GitLab после push, видит: pipeline выполняется, затем получает статус. Если есть ошибки – заходит внутрь, читает лог или скачивает отчет. Это приучает к навыкам CI/CD и отладке по логам, что само по себе является ценным профессиональным умением.

Вопросы информационной безопасности и конфиденциальности

При организации вычислительной инфраструктуры особое внимание уделяется информационной безопасности, защите учетных данных и конфиденциальности данных студентов. Ниже приведены основные положения, которые необходимо соблюдать:

Хранение токенов и ключей доступа. API-ключи для доступа к LLM-платформам (OpenAI API ключ, токен доступа Sber GigaChat, Yandex Cloud API ключ и пр.) должны храниться только в защищенных хранилищах, недоступных в репозитории в открытом виде. Категорически запрещается коммитить ключи в Git. Для локального тестирования студенты могут хранить ключи в переменных окружения или в файле конфигурации, который добавлен в `.gitignore`. В CI/CD рекомендуется использовать секретные переменные проекта: преподаватель или администратор GitLab добавляет API-токены в настройках проекта (Settings -> CI/CD -> Variables) с метками protected или masked, чтобы они не выводились в лог. В самом `.gitlab-ci.yml` вместо явных значений подставляются ссылки на эти переменные (например, `$OPENAI_API_KEY`). Такой подход гарантирует, что даже имея доступ к репозиторию, студент не видит чужих ключей, и при экспорте проекта секреты не утекут. В отчетах и логах нельзя допускать печати секретов (GitLab обычно маскирует их автоматически, если они добавлены как masked variables).

Конфиденциальность пользовательских данных. Студентам разъясняется, что любые личные данные или чувствительная информация, которую они вводят в промпты LLM, может быть сохранена на стороне внешнего сервиса. Поэтому запрещается использовать реальные персональные данные (свои или тем более чужие) в формулировках промптов. Все кейсы должны носить учебный, обезличенный характер. Если по заданию требуется обработать датасет, содержащий персональные сведения, такие данные должны быть либо синтетическими, либо обезличенными. Кроме того, OpenAI и другие сервисы используют введенные пользователями данные для обучения моделей (если не отключить эту опцию). В связи с этим, любые чувствительные фрагменты (например, фрагменты закрытого кода компании, коммерческая тайна и пр.) не должны быть отправлены в публичные LLM. В рамках учебного курса мы работаем с открытой информацией.

Правила использования сторонних сервисов. Следует придерживаться пользовательских соглашений соответствующих платформ. Например, у OpenAI есть политика содержания, у Яндекса – ограничения на использование их API (количество запросов в секунду и пр.). Студенты должны быть проинформированы о том, что несанкционированное использование API (вирусная нагрузка, попытки обхода ограничений) недопустимо. Также, если используются бесплатные версии с ограничениями, не следует создавать избыточные нагрузки – это может привести к временной блокировке доступа по IP или аккаунту.

Запрет на использование VPN и прокси для обхода блокировок. В соответствии с политикой учебного заведения, а также во избежание нарушения лицензионных

соглашений, запрещено использование VPN или анонимайзеров для доступа к сервисам, которые официально недоступны. В контексте данного курса это означает, что если с кампусной сети напрямую не открывается ChatGPT (OpenAI) или иной внешний ресурс, использовать его через VPN нельзя. Вместо этого предложены альтернативы (GigaChat, YandexGPT, DeepSeek), свободно работающие из России. Это правило продиктовано как юридическими соображениями (соблюдение законов о санкциях и ограничениях), так и вопросами безопасности (нежелательно, чтобы учебный трафик проходил через непроверенные VPN-сервисы, которые могут компрометировать данные).

Безопасность инфраструктуры. GitLab-платформа должна быть настроена с учетом защиты данных: доступ к группам проектов – по приглашениям; права гостей или репортеров для посторонних – отключены; все проекты студентов приватны. Если используется собственный сервер GitLab, он должен быть обновлен до актуальной версии и находиться под защитой (SSL, фаервол). Резервное копирование репозиториев – рекомендуется (на случай сбоев, чтобы не потерять результаты работ). Данные, генерируемые во время CI (артефакты, отчеты), хранятся ограниченное время (по умолчанию 7-30 дней) – этого достаточно для проверки, старые артефакты автоматически удаляются, сокращая риск утечки информации.

Этичное использование AI. Последний аспект безопасности – это этика и академическая честность. Студентам объясняется, что LLM – инструмент для помощи, но не средство для плагиата. Использование готовых решений из AI для прохождения контроля знаний недопустимо (если только это не предусмотрено заданием). В методических указаниях стоит подчеркнуть, что все обращения к AI должны быть осмысленными и служить образовательным целям.

Соблюдение всех перечисленных мер гарантирует, что вычислительная инфраструктура будет не только функциональной, но и безопасной для всех участников образовательного процесса.

Чек-лист корректности реализации инфраструктуры

Для проверки того, что инфраструктура развёрнута и настроена правильно, используйте следующий чек-лист. Он поможет убедиться, что все компоненты работают согласованно и отвечают требованиям:

Рабочие места готовы: На компьютерах студентов и преподавателя установлены требуемые ОС и обновленные браузеры. Все необходимое ПО (Git, Docker, Python, редакторы) инсталлировано и протестировано в работе. Интернет-соединение стабильное, сайты LLM-платформ открываются (без VPN).

GitLab и репозитории: Создана группа проектов курса на GitLab (например, `prompt-engineering-2025`). Шаблонный репозиторий (или несколько, по числу лабораторных) подготовлен и содержит: структуру папок, примеры, `.gitlab-ci.yml`, автотесты, инструкции. Каждому студенту выдан личный репозиторий (форк или проект из шаблона) и доступ к нему. Права доступа настроены (студент – Developer своего проекта, преподаватель – Maintainer всех проектов). Проверено, что коммиты пушатся и видны в удаленном репо.

CI/CD конвейер: Файл `.gitlab-ci.yml` валидный (GitLab не выдает ошибок синтаксиса). GitLab Runner зарегистрирован и имеет тег, указанный в конфигурации CI (если используется свой Runner). При пуше в репозиторий пайплайн автоматически стартует: это можно проверить, сделав тестовый коммит. Runner успешно подтягивает

нужный Docker-образ и выполняет задачи. В случае использования общего раннера – убедиться, что проект имеет достаточный лимит минут или включен в учебную подписку.

Автотесты: Предварительный прогон автотестов прошел успешно. Например, в шаблонном репозитории можно оставить реализацию по умолчанию, которая заведомо проходит тесты (или отдельный “пробный” тест, всегда проходящий). Убедиться, что зеленый pipeline получается в эталонном решении. Если автотесты заведомо должны падать на заготовке (что тоже возможно, чтобы стимулировать решение задачи), то это оговорено, и красный статус в начале – это норма. В любом случае, важно проверить, что сами тесты корректно выполняются и при допущении ошибок выводят понятные сообщения.

Отчеты и артефакты: Настроено сохранение артефактов CI: JUnit-отчет (`report.xml` или аналогичный) и/или HTML-отчет. В настройках `.gitlab-ci.yml` указаны пути к артефактам и срок их хранения. После прогона тестового pipeline видно, что в разделе Jobs > Artifacts присутствуют файлы отчетов, их можно загрузить и просмотреть локально. При открытии HTML-отчета в браузере отображается структура результатов (если применяется HTML). В интерфейсе GitLab на вкладке Pipeline > Test Report отображаются названия тестов и их статус – значит, парсинг отчетов удался.

LLM-платформы: Проверено, что все 4 рекомендованные платформы доступны и функционируют:

– ChatGPT: учетная запись OpenAI создана; при обращении через веб-интерфейс запросы на английском и русском выдают ответы. (Если доступ ограничен, принято решение, как быть: либо использовать вне аудитории, либо пропустить этот инструмент.) Если планируется API – тестовый вызов через curl или Postman с API-ключом возвращает 200 OK и результат.

– DeepSeek: веб-чат открывается, модель отвечает на тестовый вопрос. Регистрируется аккаунт для API (при необходимости) и выполняется пробный API-запрос.

– GigaChat: выполнен вход через Сбер ID на giga.chat, чат работает (например, задан вопрос на русском – получен ответ). Через кабинет разработчика Sber проверено получение API-ключа, выполнен тест cURL запрос к эндпойнту GigaChat API с полученным токеном (например, метод completions) – получен ответ от модели.

– YandexGPT: проверена работа Алисы с сложным вопросом (убеждаемся, что отвечает именно новая модель). Если используется Yandex Cloud API – получен или сгенерирован API-ключ, выполнен тестовый вызов к endpoint’у Яндекса (например, с помощью `curl` к https://api.cloud.yandex.net/ai/studio/v1/… с заданным промптом) – получен ответ.

Безопасность: Убедиться, что никакие секреты не закоммичены. Просмотреть репозитории (особенно `.gitlab-ci.yml` и код тестов) на отсутствие явных ключей, паролей. Проверить, что в GitLab CI Variables добавлены необходимые токены (`OPENAI_API_KEY`, `GIGACHAT_CLIENT_ID`/`SECRET`, и т.д.) и что они помечены как protected/masked. Попробовать запустить pipeline с использованием этих переменных и убедиться, что в логах не светятся их значения.

Отказ от VPN: На тестовом рабочем месте, подключенном к той же сети, что и будут студенты, попробовать обратиться ко всем сервисам. Проверить, что без VPN открываются DeepSeek, GigaChat, YandexGPT. В случае с ChatGPT – либо тоже открыт, либо недоступен.

Если недоступен – убедиться, что эта информация донесена до студентов и в методических указаниях указано использовать альтернативы. Никаких настроек VPN-клиентов на машинах не выполнено (или они отключены).

Резервный канал: Опционально, предусмотреть резервный вариант на случай сбоев: например, если GitLab SaaS ляжет, иметь локальный Git-сервер для офлайн сдачи, или если вдруг внешние LLM API будут недоступны, иметь локальную копию модели (типа Llama 2 7B) для демонстрации. Это не строго необходимо, но как пункт проверки – полезно продумать.

План-проспект методических указаний по организации лабораторных работ по дисциплине "Промпт инжиниринг в профессиональной деятельности"

1. Общие сведения

Образовательная программа: Искусственный интеллект и аналитика данных
Дисциплина: " Промпт инжиниринг в профессиональной деятельности "
Вид обеспечения: Проведение лабораторных работ
Условия применения:

- Курс рассчитан на студентов 1-го года обучения.
- Наличие доступа к вычислительным ресурсам (GPU/CPU).
- Использование открытых датасетов и библиотек, доступ к БЯМ.

2. Цели, задачи и ожидаемые результаты

Цели:

- Закрепление теоретических знаний на практике.
- Развитие навыков работы с Промпт запросами к БЯМ в профессиональной деятельности.
- Подготовка к решению реальных задач в индустрии.

Задачи:

- Обеспечить студентов структурированными лабораторными работами.
- Предоставить доступ к необходимым вычислительным ресурсам.
- Организовать проверку и обратную связь по выполненным работам.

Ожидаемые результаты:

- Умение применять Промпт запросы к БЯМ на практике.
- Навыки работы с базовым техникам промпт-инжиниринга: шаблоны, цепочки рассуждений, system prompts.
- Опыт создания собственных промптов и анализу их эффективности.

3. Порядок реализации

3.1. Задача №1: Подготовка лабораторных работ

1) Определение тем:

- Введение в генеративный ИИ и LLM
- Архитектуры генеративных моделей
- Принципы промпт-инжиниринга
- Шаблоны и стратегии промптов

- Создание пайплайнов взаимодействия с ИИ
- Интеграция промптов в бизнес-кейсы
- Отладка, тестирование и метрики качества
- Мультимодальные промпты и интерфейсы
- Этические аспекты и защита от генеративных рисков
- Курсовая работа и защита

2) Разработка заданий:

- Пошаговые инструкции.
- Примеры кода.
- Контрольные вопросы.

Разработка заданий для лабораторной работы «Принципы промпт-инжиниринга».

Название: Влияние параметров генерации и структуры промпта на результат
Цель: Научиться проектировать эффективные промпты и понимать влияние параметров генерации на выход LLM.

Задачи:

1. Сравнить генерацию текста при использовании разных структур промпта (вопрос, инструкция, ролевая модель).
2. Изменить параметры генерации (temperature, top_p, max_tokens) и проанализировать результат.
3. Сделать выводы о принципах точного управления генерацией.

Ожидаемые результаты:

- Умение формировать промпты разных типов.
- Понимание ключевых параметров генерации.
- Навык анализа влияния параметров на результат.

Инструменты и библиотеки:

- DeepSeek, ChatGPT, Perplexity, Grok / OpenAI Playground
- Python (опционально, через OpenAI API)
- Jupyter Notebook / Google Colab

Исходные данные: Примеры тем: «Объясни, что такое производная», «Напиши письмо преподавателю», «Составь список рекомендаций»

Ход работы:

1. Выберите 3 темы (см. выше).
2. Для каждой темы сформулируйте по 3 промпта:
   - В виде вопроса
   - В виде инструкции
   - В виде диалога (роль: преподаватель)
3. Сгенерируйте текст при следующих параметрах:
   - Температура: 0.2, 0.7, 1.0
   - Top_p: 1.0, 0.9
   - Max tokens: 50, 100
4. Оформите таблицу с результатами.
5. Проанализируйте:
   - Какие типы промптов дают наиболее точный результат?
   - Как влияет температура на вариативность?
   - Как изменяется стиль и длина при изменении max_tokens?

Пример промпта:

```
Ты — преподаватель по математике. Объясни студенту, что такое производная,
простым языком.
```

Пример кода (опционально):

```
import openai
openai.api_key = 'your-key'
response = openai.ChatCompletion.create(
  model="gpt-3.5-turbo",
  messages=[{"role": "user", "content": "Объясни, что такое производная"}],
  temperature=0.7,
  top_p=1.0,
  max_tokens=100
)
print(response['choices'][0]['message']['content'])
```

Анализ результатов:

- Температура 0.2 даёт шаблонный, формальный ответ.
- Температура 1.0 — более креативный и иногда некорректный результат.
- Ролевая модель даёт более понятный и дружелюбный ответ.

Требования к отчёту:

- Таблица с параметрами и результатами генерации
- Примеры промптов (минимум 3 темы × 3 структуры)
- Аналитическая часть с выводами
- Скриншоты или вывод кода

Критерии оценки (зачтено/незачтено):

Зачтено: все типы промптов реализованы, минимум 6 генераций, таблица оформлена, выводы осмысленны.

Незачтено: отсутствуют 2+ варианта промпта, нет анализа, неполный отчёт.

Формируемые компетенции:

- ML-1 П Способен применять знания об истории развития и трендах современного ИИ для формулирования корректных постановок задач и поиска перспективных способов решения проблем с помощью ИИ.
- ML-2 П Способен применять фундаментальные принципы и методы машинного обучения включая подготовку данных оценку качества моделей и работу с признаками
- DL-2 БСпособен применять и (или) разрабатывать современные архитектуры генеративных глубоких сетей.
- LLM-5 П Организует взаимодействие с генеративными моделями через проектирование, анализ и применение промптов.

5. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины (модуля)

5.1 Основная литература:

1. Чернышев, С. А. Основы программирования на Python : учебное пособие для вузов / С. А. Чернышев. — Москва : Издательство Юрайт, 2022. — 286 с. — (Высшее образование). — ISBN 978-5-534-14350-8. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/496893. (дата обращения: 19.07.2025).
2. Платонов, А. В. Машинное обучение : учебное пособие для вузов / А. В. Платонов. — Москва : Издательство Юрайт, 2022. — 85 с. — (Высшее образование). — ISBN 978-5-534-15561-7. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/508804 (дата обращения: 19.07.2025).

3. Рабчевский, А. Н. Синтетические данные и развитие нейросетевых технологий : учебное пособие для вузов / А. Н. Рабчевский. — Москва : Издательство Юрайт, 2024. — 187 с. — (Высшее образование). — ISBN 978-5-534-17716-9. — Текст : электронный // Образовательная платформа Юрайт [сайт]. — URL: https://urait.ru/bcode/545036 (дата обращения: 19.07.2025).
4. Елисеев А. И., Минин Ю. В. Разработка программных интерфейсов веб-приложений с использованием фреймворка FastAPI : учебное пособие. Тамбов: ТГТУ, 2024. 81 с. https://e.lanbook.com/book/472310 (дата обращения: 19.07.2025).
5. Лиманова Н. И. Разработка интеллектуальных чат-ботов : учебное пособие. Самара: ПГУТИ, 2024. https://e.lanbook.com/book/463568 (дата обращения: 19.07.2025).

5.2 Дополнительная литература:

1. Ф. Джеймс, Т. Майк Промт-инжиниринг для GenAI. Паттерны надежных запросов для качественных результатов, Sprint Book, 2025, 432 с.
2. А.А. Костин Промпт-инжиниринг. Язык будущего. 2025. Ridero, 594 с.
3. В.А. Петров, А.Н. Тихонов. Искусственный интеллект и обработка естественного языка.
4. Златопольский Д. М. Основы программирования на языке Python. 2-е изд. Москва: ДМК Пресс, 2018.
5. А.А. Кузнецов. Машинное обучение для обработки естественного языка.
6. Руководство Google по промпт-инжинирингу. Часть 1: основы и базовые техники https://habr.com/ru/articles/901426/
7. Документация облачных сервисов cloud.ru, yandex.cloud.
8. https://education.yandex.ru/handbook/prompting
9. https://yandex.cloud/ru/training/training-pro#Data
10. Elshall AS and Badir A (2025) Balancing AI-assisted learning and traditional assessment: the FACT assessment in environmental data science education. Front. Educ. 10:1596462. doi: 10.3389/feduc.2025.1596462 (Статья о рисках снижения базовых навыков при чрезмерной опоре на ИИ).
11. Wharton Knowledge “Without Guardrails, Generative AI Can Harm Education” (2024) https://knowledge.wharton.upenn.edu/article/without-guardrails-generative-ai-can-harm-education/#:~:text=practice%20session%2C%20yet%20scored%20about,exam%20as%20the%20control%20group Статья– эксперимент о влиянии GPT на обучение: зависимость от ИИ снижает глубокое усвоение материала.
12. Kadurin, Artur, et al. "The cornucopia of meaningful leads: Applying deep adversarial autoencoders for new molecule development in oncology." Oncotarget 8.7 (2016): 10883.
13. Kadurin, Artur, et al. "druGAN: an advanced generative adversarial autoencoder model for de novo generation of new molecules with desired molecular properties in silico." Molecular pharmaceutics 14.9 (2017): 3098-3104.
14. Polykovskiy, Daniil, et al. "Molecular sets (MOSES): a benchmarking platform for molecular generation models." Frontiers in pharmacology 11 (2020): 565644.
15. Khrabrov, Kuzma, et al. "$\nabla^ 2$ DFT: A Universal Quantum Chemistry Dataset of Drug-Like Molecules and a Benchmark for Neural Network Potentials." Advances in Neural Information Processing Systems 37 (2024): 36869-36889.
16. Polykovskiy, Daniil, et al. "Entangled conditional adversarial autoencoder for de novo drug discovery." Molecular pharmaceutics 15.10 (2018): 4398-4405.

17. Николенко, Сергей, Кадурин, Артур и Архангельская Екатерина. Глубокое обучение. " Издательский дом"" Питер""", 2017.
18. https://openreview.net/forum?id=FMMF1a9ifL
19. https://openreview.net/forum?id=ElUrNM9U8c#discussion
20. https://openreview.net/forum?id=JoO6mtCLHD
21. https://aclanthology.org/2024.findings-emnlp.760/
22. https://aclanthology.org/2020.coling-main.588/
23. https://link.springer.com/chapter/10.1007/978-3-030-72113-8_30
24. https://link.springer.com/chapter/10.1007/978-3-031-42448-9_10
25. https://aclanthology.org/2024.findings-naacl.288/
26. Sun, X., Li, J., Kovalenko, A.V., Feng, W., Ou, Y. Integrating Reinforcement Learning and Learning From Demonstrations to Learn Nonprehensile Manipulation //IEEE Transactions on Automation Science and Engineering, 2023, 20(3), 1735–1744, DOI: 10.1109/TASE.2022.3185071, Q1
27. Petukhova, A.V.; Kovalenko, A.V.; Ovsyannikova, A.V. Algorithm for Optimization of Inverse Problem Modeling in Fuzzy Cognitive Maps. Mathematics 2022, 10, 3452. DOI: 10.3390/math10193452, Q1
28. Kirillova, E.; Kovalenko, A.; Urtenov, M. Study of the Current–Voltage Characteristics of Membrane Systems Using Neural Networks. AppliedMath 2025, 5, 10. https://doi.org/10.3390/appliedmath5010010,

5.3. Периодические издания:

1. Базы данных компании «Ист Вью» http://dlib.eastview.com
2. Электронная библиотека GREBENNIKON.RU https://grebennikon.ru/

5.4. Интернет-ресурсы, в том числе современные профессиональные базы данных и информационные справочные системы

Электронно-библиотечные системы (ЭБС):

1. ЭБС «ЮРАЙТ» https://urait.ru/
2. ЭБС «УНИВЕРСИТЕТСКАЯ БИБЛИОТЕКА ОНЛАЙН» http://www.biblioclub.ru/
3. ЭБС «BOOK.ru» https://www.book.ru
4. ЭБС «ZNANIUM.COM» www.znanium.com
5. ЭБС «ЛАНЬ» https://e.lanbook.com

Профессиональные базы данных

1. Scopus http://www.scopus.com/
2. ScienceDirect https://www.sciencedirect.com/
3. Журналы издательства Wiley https://onlinelibrary.wiley.com/
4. Научная электронная библиотека (НЭБ) http://www.elibrary.ru/
5. Полнотекстовые архивы ведущих западных научных журналов на Российской платформе научных журналов НЭИКОН http://archive.neicon.ru
6. Национальная электронная библиотека (доступ к Электронной библиотеке диссертаций Российской государственной библиотеки (РГБ) https://rusneb.ru/
7. Президентская библиотека им. Б.Н. Ельцина https://www.prlib.ru/
8. База данных CSD Кембриджского центра кристаллографических данных (CCDC) https://www.ccdc.cam.ac.uk/structures/
9. Springer Journals: https://link.springer.com/
10. Springer Journals Archive: https://link.springer.com/
11. Nature Journals: https://www.nature.com/
12. Springer Nature Protocols and Methods: https://experiments.springernature.com/sources/springer-protocols

13. Springer Materials: http://materials.springer.com/
14. Nano Database: https://nano.nature.com/
15. Springer eBooks (i.e. 2020 eBook collections): https://link.springer.com/
16. "Лекториум ТВ" http://www.lektorium.tv/
17. Университетская информационная система РОССИЯ http://uisrussia.msu.ru

Бесплатные образовательные ресурсы

1. Jupyter Notebook – интерактивные вычисления
2. Visual Studio Code – редактор кода с поддержкой Python
3. Google Scholar/arXiv – доступ к научным публикациям

Ресурсы свободного доступа

1. Промпт и Промпт инжиниринг https://courses.sberuniversity.ru/generative_art/img/13
2. Principled Instructions Are All You Need for Questioning LLaMA-1/2, GPT-3.5/4 (https://arxiv.org/abs/2312.16171).
3. Nathan Hunter. The Art of Prompt Engineering with ChatGPT
4. ChatGPT Prompt Engineering for Developers https://www.promptingguide.ai/
5. КиберЛенинка http://cyberleninka.ru/;
6. Американская патентная база данных http://www.uspto.gov/patft/
7. Министерство науки и высшего образования Российской Федерации https://www.minobrnauki.gov.ru/;
8. Федеральный портал "Российское образование" http://www.edu.ru/;
9. Информационная система "Единое окно доступа к образовательным ресурсам" http://window.edu.ru/;
10. Единая коллекция цифровых образовательных ресурсов http://school-collection.edu.ru/ .
11. Проект Государственного института русского языка имени А.С. Пушкина "Образование на русском" https://pushkininstitute.ru/;
12. Справочно-информационный портал "Русский язык" http://gramota.ru/;
13. Служба тематических толковых словарей http://www.glossary.ru/;
14. Словари и энциклопедии http://dic.academic.ru/;
15. Образовательный портал "Учеба" http://www.ucheba.com/;
16. Законопроект "Об образовании в Российской Федерации". Вопросы и ответы http://xn--273--84d1f.xn--p1ai/voprosy_i_otvety

Собственные электронные образовательные и информационные ресурсы КубГУ

1. Электронный каталог Научной библиотеки КубГУ http://megapro.kubsu.ru/MegaPro/Web
2. Электронная библиотека трудов ученых КубГУ http://megapro.kubsu.ru/MegaPro/UserEntry?Action=ToDb&idb=6
3. Среда модульного динамического обучения http://moodle.kubsu.ru
4. База учебных планов, учебно-методических комплексов, публикаций и конференций http://infoneeds.kubsu.ru/
5. Библиотека информационных ресурсов кафедры информационных образовательных технологий http://mschool.kubsu.ru;
6. Электронный архив документов КубГУ http://docspace.kubsu.ru/
7. Электронные образовательные ресурсы кафедры информационных систем и технологий в образовании КубГУ и научно-методического журнала "ШКОЛЬНЫЕ ГОДЫ" http://icdau.kubsu.ru/

6. Методические указания для обучающихся по освоению дисциплины (модуля)

По дисциплине «Промпт-инжиниринг в профессиональной деятельности» предусмотрено проведение лекционных занятий, на которых даётся систематизированное представление о генеративных моделях искусственного интеллекта (ИИ) и их применении в профессиональной деятельности. В ходе лекций студенты знакомятся с базовыми концепциями генеративного ИИ (LLM, diffusion-модели), принципами их архитектуры (Transformer, GPT, BERT, T5), методами взаимодействия с такими моделями через текстовые запросы (prompt engineering). Особое внимание уделяется анализу стратегий промпт-инжиниринга: zero-shot, few-shot, chain-of-thought prompting, а также современным практикам создания эффективных запросов к LLM. Демонстрируются практические примеры использования генеративных моделей в бизнесе, образовании, программировании, креативных индустриях. Обсуждаются ключевые риски генерации: галлюцинации, этические ограничения, защита персональных данных. После каждой лекции студентам рекомендуется выполнить прикладные задания — анализировать ответы LLM, экспериментировать с параметрами генерации и шаблонами промптов.

Лабораторные занятия посвящены формированию практических навыков работы с генеративными ИИ-моделями. Студенты последовательно осваивают методы промпт-инжиниринга на платформе ChatGPT, а также через API-интерфейсы (OpenAI, Hugging Face, YandexGPT и др.). Выполняются упражнения по настройке параметров генерации (temperature, top-p, max tokens), составлению шаблонов промптов для типовых задач (написание писем, объяснение кода, генерация описаний). Рассматриваются веб-технологии интеграции генеративных моделей: создание сервисов на FastAPI, запуск интерфейсов через Gradio или Streamlit. В ходе лабораторных занятий студенты также изучают методы тестирования и отладки промптов, проводят A/B-тестирование, оценивают качество генерации по метрикам (BLEU, ROUGE). Выполняются проекты по генерации текста, изображений и мультимодальных ответов. После каждого занятия выдаются задания для самостоятельной доработки — настройка собственных промптов, реализация цепочек взаимодействия с ИИ, анализ ответов модели.

Самостоятельная работа обучающихся направлена на углублённое изучение методологических и прикладных аспектов промпт-инжиниринга. Рекомендуется регулярно обращаться к учебникам, документации API и статьям по теме (включая публикации на arXiv, OpenAI Research, Google DeepMind). Студенты должны научиться формулировать прикладные задачи (автоматизация переписки, поддержка пользователей, генерация кода), разрабатывать промпты и пайплайны генерации, настраивать параметры вывода и оценивать релевантность результатов. Особое внимание уделяется разработке комплексных решений — от проектирования промптов до внедрения в виде web-интерфейса или CLI-инструмента. Умения в области Python-программирования и баз данных активно применяются при обработке входных/выходных данных и построении пользовательских интерфейсов.

Итоговой формой освоения курса является проектная (курсовая) работа. В её рамках студент разрабатывает полнофункциональный AI-инструмент, использующий промпты и генеративные модели. Это может быть чат-бот, генератор маркетинговых текстов, ассистент преподавателя, визуализатор знаний и др. Курсовой проект охватывает все ключевые элементы: анализ предметной области, построение промптов, тестирование и отладка, реализация пайплайна взаимодействия, анализ рисков и ограничений. Такой подход позволяет студенту проявить творческую инициативу и продемонстрировать уровень сформированных компетенций.

Для студентов с ограниченными возможностями здоровья предусмотрены индивидуальные консультации и адаптированные материалы. Преподаватель помогает осваивать интерфейсы взаимодействия с ИИ, объясняет ключевые понятия в доступной форме, предоставляет инструкции с альтернативным форматированием. При необходимости используются голосовые интерфейсы, увеличенный масштаб экрана, сопровождение при выполнении заданий. Индивидуальный подход обеспечивает равные

условия участия в образовательном процессе и достижения запланированных результатов обучения.

Рассмотрим примеры кейсов.

Подход, определяющий установление соответствия кейсов ИП и УГТ (5), позволяет четко соотносить этапы развития технологии с вовлеченностью партнера и снижать риски при переходе от лабораторных испытаний к промышленному внедрению.

УГТ5 - Проверка данного уровня проводится в средах имитационного моделирования эмулирующих условия приближенные к реальности. Таким образом реализуется основная цель, продемонстрировать уровень готовности технологии на модельной среде максимально приближенной к реальности, а также проверить соответствие технологии требованиям к производительности (провести профилирование на реальных объемах и убедиться в эффективности процедуры масштабирования).

Кейс 1. Генеративный ИИ для автоматического составления инвестиционных обзоров

Описание: Аналитики банка ежедневно готовят десятки аналитических и инвестиционных обзоров по рынкам, компаниям, макроэкономическим показателям. Задача – исследовать применение больших языковых моделей (LLM) для генерации кратких сводок и аналитических отчётов на основе входных данных (биржевые котировки, макроэкономические показатели, рыночные события). По сути, требуется автоматизировать подготовку инвестиционного обзора, чтобы экономить время аналитиков и обеспечить единообразие стиля.

Цель: Разработать инструмент на базе LLM, способный по структурированным данным и краткому описанию формировать инвестиционный обзор в деловом банковском стиле. Иными словами, настроить промпт и модель так, чтобы на основе табличных данных и краткой справки генерировался связный текст отчёта.

Ожидаемый результат: Прототип модели, генерирующей аналитический текст обзора объёмом 500–1000 слов с разделами «Обзор событий», «Рекомендации», «Прогнозы», оформленный по стандартам банка. Студент должен представить пример сгенерированного обзора и показать, как промпт-инжиниринг позволил получить осмысленный структурированный отчёт.

Кейс 2. NLP-анализ жалоб клиентов в свободной форме

Описание: В клиентскую службу банка поступают обращения от клиентов через чат, мобильное приложение и форму жалоб. Эти сообщения пишутся в свободной форме и могут содержать описания проблем, эмоции клиентов и т.д. Необходимо построить NLP-модель семантического анализа, которая с помощью методов промпт-инжиниринга выделяет суть обращения (основную проблему клиента), определяет тональность (эмоциональную окраску) и оценивает потенциальную серьёзность инцидента. Это поможет банку быстрее реагировать на проблемы и выявлять типовые болевые точки.

Цель: Автоматизировать первичную классификацию клиентских обращений для ускорения их маршрутизации и выявления повторяющихся проблем в продуктах или процессах. С помощью LLM планируется распознавать ключевые темы жалобы (например, сбой приложения, ошибка платежа) и эмоцию клиента, чтобы решать вопрос приоритетности ответа.

Ожидаемый результат: Прототип решения (например, на базе LLM или специализированной модели), который автоматически извлекает из текста обращения тему жалобы (например, «двойное списание», «не работает карта»), определяет эмоциональную тональность (негативная, нейтральная, позитивная) и степень критичности. Студент предоставит примеры анализа нескольких жалоб и покажет, как промпты настроены для извлечения этих сведений.

Кейс 3. Мультимодальный ассистент для банковского отделения

Описание: В физических отделениях банка планируется внедрение интерактивных ИИ-консультантов для помощи клиентам. Предполагается создать прототип мультимодального ассистента, который умеет воспринимать речь и визуально ориентироваться в отделении. Например, такой ассистент должен распознавать лица клиентов или предъявленные ими документы, видеть расположение банкоматов и при этом понимать голосовые запросы. Он комбинирует компьютерное зрение и обработку естественного языка, отвечая голосом на вопросы клиентов и давая визуальные подсказки.

Цель: Разработать прототип системы, имитирующей функциональность такого помощника. Это включает: ответ на типовые вопросы клиентов голосом, распознавание и описание визуальных объектов (например, отсканировать QR-код паспорта или показать путь до нужного окна), навигацию клиента внутри отделения и т.д.. Промпт-инжиниринг здесь нужен для координации разных модальностей (текстовой и визуальной) и формата ответов ассистента.

Ожидаемый результат: Интерактивная мультимодальная модель (или эмуляция её работы), которая на запрос пользователя голосом может выполнить действие: показать нужный участок отделения на экране, прокомментировать прогресс обслуживания, указать на ближайший банкомат и т.п.. Студент демонстрирует работу ассистента на нескольких сценариях (например, клиент спрашивает голосом, ассистент реагирует словами и отображает информацию на экране).

Кейс 4. Оптимизация промптов и оценка их качества в банковском чат-боте

Описание: Банк внедряет AI-чат-бот для консультирования клиентов по продуктам и услугам. Качество ответов такого бота сильно зависит от того, как сформулированы системные и пользовательские промпты (инструкции для LLM). В этом кейсе студентам предлагается проанализировать, как различные формулировки промпта влияют на ответы модели, и выработать методику оценки качества промптов. Например, короткий неопределённый запрос может приводить к расплывчатому ответу, тогда как структурированный промпт с указанием ролей и формата ответа – к более точному и полезному результату. Необходимо уметь измерять качество ответов модели (наличие ошибок, полнота ответа, соответствие инструкции и ожиданиям) и на основе этого улучшать промпты.

Цель: Разработать и протестировать подход к улучшению промптов для банковского чат-бота. Студент должен подобрать метрики качества ответов (например, доля правильных решений задач пользователя, удовлетворённость пользователей ответами, отсутствие «галлюцинаций» в ответе) и на их основе сравнить несколько вариантов промптов. Цель – добиться того, чтобы доля верных и полезных ответов выросла до целевого уровня за счёт оптимизации формулировок запросов и инструкций для LLM.

Ожидаемый результат: Отчёт или мини-исследование, демонстрирующее оценку качества разных вариантов промптов и финальную оптимизированную версию. Например, студент может привести таблицу с метриками (релевантность ответа, фактическая корректность, соблюдение стиля и т.д.) для промпта версии A vs версии B, и показать, что после улучшения промпта качество ответов повысилось. Ожидается, что студент предложит рекомендации по формированию эффективных промптов (структурирование инструкции, указание контекста, примеров и пр.) на основе проведённых экспериментов.

Кейс 5. LLM + RAG как интерфейс к BI-системе (строительная компания)

Описание: Для корпоративной BI-системы (Business Intelligence) компании-застройщика необходимо разработать интерфейс на естественном языке, чтобы сотрудники могли запрашивать отчёты и данные без знаний SQL. Проект предполагает интеграцию

LLM с хранилищем данных (например, ClickHouse) по принципу Retrieval-Augmented Generation (RAG). LLM будет классифицировать естественно-языковые запросы пользователей и извлекать из них фактические параметры (фильтры, метрики, периоды), необходимые для построения отчёта. Затем эти параметры передаются BI-сервису для формирования ответа. Таким образом, пользователь задаёт вопрос на обычном языке, а система формирует корректный запрос к данным.

Цель: Разработать промпт-стратегии и логику обработки запросов, позволяющие LLM преобразовывать вопрос пользователя в параметры для типового отчёта. Например, на запрос «Покажи продажи квартир по регионам за последний год» модель должна извлечь сущности: показатель – продажи квартир, измерение – регионы, период – последний год, и выбрать соответствующий шаблон отчёта. Цель – упростить доступ к данным с помощью промптов, чтобы даже неспециалисты могли получать BI-отчёты.

Ожидаемый результат: Прототип инструмента на основе LLM, который понимает запросы о данных и формирует нужный отчёт или визуализацию. Студент демонстрирует несколько примеров запросов (о продажах, затратах, прогрессе строительства и т.д.) и полученных результатов. Оценивается, насколько правильно LLM интерпретирует разные формулировки вопросов и извлекает нужные параметры – т.е. фактически качество промпт-интеграции с BI.

Кейс 6. Генеративный ИИ для создания проектной документации по техническому заданию

Описание: В девелоперской (строительной) компании архитекторы и инженеры тратят значительное время на подготовку текстовой проектной документации: обоснования принятых решений, пояснительные записки, описания инженерных систем и др. Задача – исследовать возможность использования LLM для генерации черновиков таких документов на основе исходных данных проекта. Входные данные могут включать: параметры здания (этажность, материалы), климатические условия, назначение объекта, нормативные требования и т.п.. Требуется настроить промпты так, чтобы модель на основе этих структурированных входных сведений генерировала связный текст, соответствующий требованиям ГОСТ и принятым шаблонам в компании.

Цель: Разработать прототип текстового генератора, помогающего специалистам быстрее формировать проектную документацию согласно заданным шаблонам и нормам. Проще говоря, цель – автоматизировать рутинную часть написания документации: LLM предлагает черновой текст, а инженер уже его дорабатывает. Для этого нужно продумать, как представить исходные данные в промпте (например, перечислить основные параметры и требования) и как ограничить стиль и содержание генерируемого текста рамками технического задания.

Ожидаемый результат: Инструмент или модель на основе LLM, генерирующая логически стройный и нормативно корректный текст проектного документа. Студент предоставляет фрагменты сгенерированных документов (например, раздел пояснительной записки), показывая, что текст содержит обоснования решений, описания систем и пр. Ожидается, что такой черновик будет легко поддаваться правке инженером, а применение промпт-инжиниринга сократит время подготовки документации.

Кейс 7. Мультимодальный агент для мониторинга строительной площадки

Описание: Компания-застройщик хочет создать систему для мониторинга хода строительства с помощью ИИ. Предлагается прототип мультимодального агента, способного анализировать визуальные данные (фото и видео со стройплощадки) и понимать голосовые или текстовые запросы инженеров. Например, специалист может спросить: «Проверь, выполнен ли монтаж перекрытия на 5 этаже?» – агент должен проанализировать последние фотографии 5-го этажа и ответить на основе увиденного. Таким образом, объединяются возможности компьютерного зрения (распознавание объектов, стадий строительства, выявление нарушений) и NLP (понимание запроса, генерация отчёта).

Цель: Объединить в одном решении анализ изображений и понимание естественного языка для контроля за строительством. Необходимо настроить модель так, чтобы по запросу она находила нужный визуальный фрагмент (например, снимок определённого этажа), оценила его (выявила, что сделано, а что нет) и сформулировала ответ для пользователя. Цель проекта – доказать, что мультимодальный подход может ускорить проверку прогресса работ и обнаружение проблем на объекте.

Ожидаемый результат: Прототип интерактивного агента, который по запросу специалиста показывает нужный участок стройки, комментирует статус (например: «Перекрытие на 5 этаже установлено на 80%») и фиксирует обнаруженные нарушения. Студент демонстрирует работу агента на нескольких запросах. Оценивается, насколько корректно агент распознаёт ситуацию на изображениях и понятно ли объясняет её в тексте – то есть как промпт-инжинированием связаны визуальные данные и генерация ответа.

Кейс 8. Генерация маркетингового контента для жилых комплексов (текст + изображения)

Описание: Девелоперская группа регулярно запускает маркетинговые кампании для своих жилых комплексов. Необходимо исследовать применение диффузионных моделей генерации изображений совместно с LLM для подготовки рекламных материалов. Задача – автоматически создавать визуализированные рекламные карточки объектов: например, генерировать изображение интерьера квартиры или вид из окна с помощью diffusion-модели, а также генерировать привлекательное текстовое описание квартиры, инфраструктуры района, преимуществ жилого комплекса с помощью LLM. Идея в том, чтобы на ранних этапах, без привлечения дизайнеров и копирайтеров, получать черновой вариант рекламного контента для тестирования идей.

Цель: Создать инструмент(ы) для быстрой генерации продающих материалов на основе заданных характеристик объекта недвижимости. Например, по вводу параметров квартиры (метраж, этаж, расположение) и окружения, система генерирует изображение этой квартиры в определённом стиле и пишет рекламный текст о ней. Цель – снизить время и стоимость подготовки маркетинговых кампаний, используя промпт-инжиниринг для получения вариативного контента.

Ожидаемый результат: Набор автоматически сгенерированных карточек недвижимости, где у каждого объекта есть сгенерированное изображение и описание, близкое по стилю к работе маркетолога. Студент представляет несколько примеров таких карточек (например, картинка + абзац текста) и оценивает их качество. Ожидается обсуждение, как подбирались промпты для моделей (например, текстовые подсказки для генерации нужного визуального стиля и содержания описания) и какие правки потребовались, чтобы довести материалы до приемлемого уровня.

Кейс 9. Прогнозирование сроков сдачи строительного объекта по текстовым и визуальным данным

Описание: Строительная компания ведёт архив данных о ходе строительства: регулярные отчётные записи (текстовые отчёты о строительных работах) и фотографии со стройки за разные даты. Предполагается, что с помощью мультимодели, обрабатывающей текст + изображения, можно прогнозировать вероятность задержки сдачи объекта относительно плана. То есть объединить информацию из последних отчётов и визуального прогресса, чтобы понять, укладывается ли проект в график. Модель должна выявлять признаки отставания, например, незавершённые фасадные работы или отсутствие монтажа инженерных систем, которые упоминаются в тексте и видны на фото.

Цель: Разработать модель (возможно, с использованием LLM для анализа текста отчётов и CV-модели для фото) для оценки рисков задержки сдачи объекта по текущему статусу. Промпт-инжиниринг может использоваться для превращения выводов модели (например, отмеченных визуальных несоответствий) в понятное объяснение. Цель – предупредить менеджеров проекта о потенциальных проблемах заранее, на основе комплексного анализа данных.

Ожидаемый результат: Прототип системы, показывающей вероятность отклонения от графика и сопровождающей это объяснением причин. Студент должен представить, например, интерактивный дашборд или отчёт: в котором указано, что объект №X имеет риск задержки 30%, и перечислены ключевые факторы (выявленные моделью) – например, «не завершены фасадные работы на 80% срока», «монтаж отопления не начат, хотя по плану должен быть готов на 50%» и т.д. Таким образом демонстрируется, как комбинация данных и промпт-инжиниринга позволяет получить объяснимую предиктивную аналитику.

7. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю)

7.1 Перечень информационно-коммуникационных технологий

Инструменты и библиотеки: используются современные LLM-платформы и инструменты генерации. В частности, ChatGPT, DeepSeek, Perplexity, Grok и аналогичные сервисы для взаимодействия с готовыми моделями. Для работы с моделями открытого доступа применяются Hugging Face (Spaces и Inference API). Веб-разработка и интеграция реализуются на Python (версии 3.7+) с библиотеками FastAPI, Uvicorn, requests и dotenv. Для тестирования и анализа выходных данных могут использоваться библиотеки NLP (например, nltk и rouge-score). Инструментарий также включает Google Colab и Jupyter Notebook, а для оформления отчетов – MS Word или Markdown-редактор.

Исходные данные: для экспериментов и заданий используются учебные тексты и кейсы. Это могут быть примеры заданий по математике или программированию, на основе которых формулируются промпты (например, «Объясни, что такое производная», «Напиши письмо преподавателю»). Для мультимодальных задач требуются картинки (наборы изображений) или описания к ним. Данные выбираются или генерируются преподавателем заранее.

Программное обеспечение и ИКТ: необходим доступ к интернету и API ключи (OpenAI, Hugging Face и др.). Рекомендуется использовать среды разработки Jupyter Notebook или Google Colab, а также удобный код-редактор (Visual Studio Code). Для хранения и совместной работы со студентами можно задействовать учебную информационную систему (LMS). Кроме того, для анализа результатов и визуализации могут понадобиться Excel или Google Sheets.

1. Облачные платформы и сервисы

cloud.ru,YandexCloud,AWS/GCP/Azure– облачные вычисления

2. Системы управления версиями и коллаборации

Git/GitHub/GitLab – контроль версий кода и совместная разработка

4. Система управления обучением

Moodle – сдача работ

7.2 Перечень лицензионного и свободно распространяемого программного обеспечения

1. Лицензионное ПО

VSCode – IDE для Python (свободнораспространяемое)

LibreOffice– оформление отчетов (свободнораспространяемое)

2. Свободное ПО (Open Source)

Hugging Face Transformers – предобученные модели (BERT, GPT)

Gensim – тематическое моделирование и word2vec

GitLab, GIT, MLFlow, Docker, Kubernetes, Terraform.Фреймворки для ML:

PyTorch/TensorFlow – разработка нейросетей

scikit-learn – классические алгоритмы ML

Инструменты для визуализации:

Streamlit/Gradio – создание веб-интерфейсов для моделей
Matplotlib/Seaborn – графики и анализ данных

СУБД:

SQLite/PostgreSQL – хранение структурированных данных
FAISS/Annoy – векторный поиск

8. Материально-техническое обеспечение по дисциплине (модулю)

Виртуальные машины, кластер Managed Kubernetes и ресурсы GPU в облаке предоставляется индустриальным партнером ПАО «Сбербанк»:

№	Продукт	Параметры продукта	Кол-во	Кол-во конфигураций	Ед. изм.
1	Виртуальная машина	Виртуальная машина 10% vCPU 2 vCPU 4 RAM	1	60	Шт
		ОС Ubuntu 22.04	1		Шт
		Системный диск SSD	1		Шт
			10		Гб
		Аренда публичного IP	1		Шт
2	Виртуальная машина с GPU	Виртуальная машина с GPU NVIDIA® Tesla® V100 2 GPU 8 vCPU 128 ГБ RAM	1	1	Шт
		ОС Ubuntu_24.04	1		Шт
		Системный диск SSD	1		Шт
			2000		Гб
		Диск SSD	1		Шт
			4096		Гб
		Диск SSD	1		Шт
			4096		Гб
		Аренда публичного IP	1		Шт
3	K8S	Master node 8 vCPU 16 RAM	1	1	Шт
		Worker node 10% доля 4 vCPU 32 RAM	5		Шт
		Worker node SSD-NVME	64		Гб
		Аренда публичного IP	1		Шт

4	ML Inference Instance Type GPU	Время работы в месяц	40	1	Ч
		Инстанс 8 x NVIDIA® H100 NVLink PCIe 160 vCPU 1520 GB RAM	1		Шт
		Количество запросов к ML-моделям	1		Млн. Шт
		Кэш ML-моделей	160		Гб
5	LLM	Токены GigaChat 2 Max	50		Млн. Шт
		Токены Embeddings	400		Млн. Шт

Дополнительные облачные ресурсы предоставляются технологическим партнером Yandex Cloud.

№	Вид работ	Наименование учебной аудитории, ее оснащенность оборудованием и техническими средствами обучения
1.	Лекционные занятия	Аудитория, укомплектованная специализированной мебелью и техническими средствами обучения
2.	Лабораторные занятия	Аудитория, укомплектованная специализированной мебелью и техническими средствами обучения, компьютерами, проектором, программным обеспечением
3.	Практические занятия	Аудитория, укомплектованная специализированной мебелью и техническими средствами обучения
4.	Групповые (индивидуальные) консультации	Аудитория, укомплектованная специализированной мебелью и техническими средствами обучения, компьютерами, программным обеспечением
5.	Текущий контроль, промежуточная аттестация	Аудитория, укомплектованная специализированной мебелью и техническими средствами обучения, компьютерами, программным обеспечением
6.	Самостоятельная работа	Кабинет для самостоятельной работы, оснащенный компьютерной техникой с возможностью подключения к сети «Интернет», программой экранного увеличения и обеспеченный доступом в электронную информационно-образовательную среду университета.

Примечание: Конкретизация аудиторий и их оснащение определяется ОПОП.