

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.07 Обработка естественного языка

Направление подготовки 02.03.02 Фундаментальная информатика и
информационные технологии

Профиль Современные методы машинного обучения и компьютерного зрения

Форма обучения очная

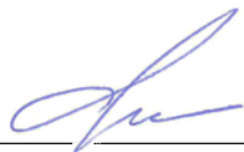
Квалификация бакалавр

Краснодар 2025

Рабочая программа дисциплины «Обработка естественного языка» составлена в соответствии с федеральным государственным образовательным стандартом высшего образования (ФГОС ВО) по направлению подготовки 02.03.02 Фундаментальная информатика и информационные технологии.

Программу составил(и):

Добровольская Н.Ю. доцент, канд. пед. наук, доцент

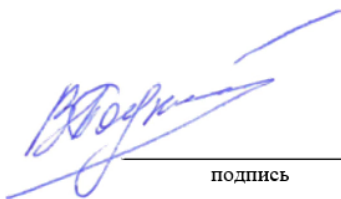


подпись

Рабочая программа обсуждена на заседании кафедры информационных технологий протокол № 1 «26» августа 2025г.

Заведующий кафедрой

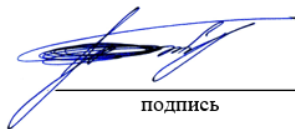
Подколзин В.В.



подпись

Утверждена на заседании учебно-методической комиссии факультета компьютерных технологий и прикладной математики протокол № 1 «28» августа 2025г.

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 и практических навыков в области обработки естественного языка (NLP) с акцентом на применение глубокого обучения, включая архитектуры LLM и трансферное обучение.

1.2 Задачи дисциплины

Изучить математические основы NLP (векторизация, вероятностные модели).

Освоить архитектуры нейросетей для NLP (RNN, Transformer, BERT, GPT).

Научиться применять предобученные модели (Hugging Face) для классификации, генерации и анализа текста.

Разрабатывать и оптимизировать NLP-пайплайны (дообучение, RAG, развертывание).

1.3 Место дисциплины (модуля) в структуре образовательной программы

Дисциплина «Обработка естественного языка» относится к «Часть, 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:

1. Проектирование и построение ETL-процессов
2. Создание и оптимизация хранилищ данных
3. Обеспечение качества и доступности данных
4. Настройка инфраструктуры для обработки больших данных
5. Интеграция разрозненных источников данных
6. Работа с данными в области природопользования, медицины, связи и телекоммуникаций

Роль 2: ML Engineer (Инженер МО)

Задачи:

1. Реализация ML-моделей в продуктивных системах
2. Оптимизация производительности и масштабирование моделей
3. Разработка ML-пайплайнов и автоматизация процессов
4. Мониторинг качества моделей в продуктиве
5. Интеграция ML-решений с бизнес-приложениями

Роль 3: MLOps (Специалист по эксплуатации ИИ)

Задачи:

6. Автоматизация процессов обучения и развертывания моделей
7. Мониторинг производительности ML-систем
8. Управление версиями моделей и данных
9. Обеспечение CI/CD для ML-проектов
10. 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:

DL-1	<i>Способен применять и (или) разрабатывать архитектуры глубоких нейронных сетей.</i>
DL-1.1	Способен объяснять и применять математические основы нейронных сетей, включая расчет градиентов, методы оптимизации и алгоритм обратного распространения ошибки (backpropagation), для эффективного обучения моделей.
	Знать: математические основы нейронных сетей, включая расчет градиентов, методы оптимизации и алгоритм обратного распространения ошибки Уметь: модифицировать популярные архитектуры графовых нейронных сетей (GCN, GraphSAGE, GIN); применяет графовые сети внимания (GAT). Адаптирует архитектуры под специфические типы графов (гетерогенные, динамические). Разрабатывает кастомные механизмы агрегации. Владеть: методами расчета градиентов и оптимизации нейронных сетей
DL-1.3	Способен применять современные архитектуры глубоких сетей для решения различных задач, понимая их внутреннюю структуру и особенности обучения.
	Знать: современные архитектуры глубоких сетей и их внутреннюю структуру Уметь: применять глубокие сети для решения различных задач. Разрабатывает кастомные архитектуры для специфических комбинаций модальностей. Оптимизирует совместное обучение нескольких задач (multi-task learning). Владеть: методами обучения современных архитектур глубоких сетей. Применяет продвинутые техники (cross-modal distillation, contrastive learning). Создает эффективные пайплайны обработки мультимодальных данных
DL-1.5	Способен разрабатывать, адаптировать и внедрять генеративные нейронные сети для решения практических задач, включая создание новых архитектур, оптимизацию обучения и промышленное развертывание моделей
	Знать: архитектуры генеративных нейронных сетей. Понимает и применяет специфические архитектурные варианты для задач (например, Encoder-Decoder для перевода/суммаризации, Decoder-only для генерации). Уметь: разрабатывать и адаптировать генеративные сети для практических задач. Использует техники для работы с длинными последовательностями: эффективные схемы внимания (LInformer, Performer, Longformer, Sparse Transformers). Проводит детальный анализ ошибок модели, понимает типичные проблемы (bias в данных/модели, переобучение) и применяет методы борьбы с ними (регуляризация, аугментация данных) Владеть: промышленным развертыванием генеративных моделей
DL-1.8	Способен разрабатывать, адаптировать и внедрять трансформерные архитектуры для решения задач обработки последовательностей, включая создание новых моделей, оптимизацию обучения и промышленное развертывание
	Знать: трансформерные архитектуры и их особенности Уметь: адаптировать трансформерные архитектуры для решения задач обработки последовательностей Владеть: промышленным развертыванием трансформерных моделей
DL-2	<i>Способен применять и (или) разрабатывать современные архитектуры генеративных глубоких сетей</i>
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

	<p>Знать: известные архитектуры генеративных глубоких нейронных сетей Уметь: применять генеративные сети для решения прикладных задач. Настраивает параметры генерации под конкретную задачу. Реализует техники контролируемой генерации (guidance scale, prompt engineering). Комбинирует несколько моделей в пайплайн. Проводит domain adaptation для специфических данных. Использует продвинутые методы эффективного обучения (QLoRA, DreamBooth). Оптимизирует процесс обучения (подбор lr, батчей). Владеть: методами дообучения генеративных моделей на наборах данных. Модифицирует архитектуры под специфические требования. Разрабатывает гибридные подходы (например, диффузионные модели + GAN). Оптимизирует архитектуры для целевых аппаратных платформ. Разрабатывает новые методы дообучения для генеративных моделей. Применяет few-shot/zero-shot learning техники. Реализует reinforcement learning для генерации.</p>
DL-2.2	Имплементирует известные архитектуры генеративных сетей, реализует пайплайны их обучения на датасетах и вывод генеративных моделей в продуктивную среду
	<p>Знать: архитектуры генеративных сетей и пайплайны их обучения Уметь: имплементировать архитектуры генеративных сетей. Разрабатывает кастомные реализации генеративных архитектур. Создает high-load генеративные сервисы. Разрабатывает системы кеширования генераций. Реализует сложные A/B тестирования. Написание кастом CUDA-ядер. Владеть: вывода генеративных моделей в продуктивную среду</p>
DL-4	<i>Способен применять и (или) разрабатывать алгоритмы, методы и технологии обработки естественного языка</i>
DL-4.1	Имплементирует известные алгоритмы, архитектуры и модели обработки естественного языка на реальных данных, строит пайплайны обучения моделей и развертывания NLP-сервисов в продуктивной среде
	<p>Знать: алгоритмы, архитектуры и модели обработки естественного языка Уметь: имплементировать NLP-модели на реальных данных. Владеет инструментами грамматического разбора структурированных и слабо-структурированных текстов, способен написать свой парсер. Владеет инструментами разметки текстовых данных и формирования словарей. Владеть: развертыванием NLP-сервисов в продуктивной среде</p>
DL-4.2	Определяет стек технологий, методов и алгоритмов для построения продуктов с обработкой естественного языка (диалоговые системы, вопросно-ответные системы, рекомендательные системы и т.д.)
	<p>Знать: основные архитектуры сетей, использующиеся для векторизации текстовых данных: Word2Vec, Doc2Vec, Glove, FastText, рекуррентные нейронные сети и сети-трансформеры (энкодеры). Уметь: применять простейшие классические модели для извлечения полезных признаков из текстов: мешок слов, TF/IDF, латентный семантический анализ. Владеть: самостоятельно находит подходящую модель для векторизации текстовых данных в открытых источниках и применить её для практической задачи.</p>
DL-4.3	Имплементирует известные алгоритмы, архитектуры и модели обработки естественного языка на реальных данных, строит пайплайны обучения моделей и развертывания NLP-сервисов в продуктивной среде
	Знать: классические алгоритмы тематического моделирования и способен применить их в практической задаче: LSA, pLSA, BigARTM.

	<p>Уметь: обучать модель классификации текстовых данных с использованием векторизованного представления текста - разреженного в случае применения классических статистических алгоритмов ML для классификации и плотного для применения в нейросетевых моделях)</p> <p>Владеть: архитектурой и применяет рекуррентные нейронные сети и сети типа "трансформер" для downstream задач, требующих языкового моделирования (контекстуализированные векторные представления, генерация текстов, аннотирование и реферирование текстов и т.п.)</p>
DL-4.4	Разрабатывает новые алгоритмы и библиотеки обработки естественного языка, новые архитектуры глубоких нейронных сетей и методы их обучения для задач анализа текста
	<p>Знать: архитектуру рекуррентных нейронных сетей и сетей типа "трансформер" для downstream задач</p> <p>Уметь: применяет рекуррентные нейронные сети и сети типа "трансформер" для downstream задач, требующих языкового моделирования (контекстуализированные векторные представления, генерация текстов, аннотирование и реферирование текстов и т.п.)</p> <p>Владеть: Использует предобученные модели без модификации архитектуры.</p>
LLM-1	Способен применять и (или) разрабатывать генеративные модели и БЯМ
LLM-1.1	Знает архитектуры генеративных моделей.
	<p>Знать: архитектуры генеративных моделей</p> <p>Уметь: различать архитектуры генеративных моделей. Сравнивает архитектуры и выбирает подходящую под задачу</p> <p>Владеть: знаниями об архитектурах генеративных моделей</p>
FC-2	Способен проводить фронтирные исследования в области фундаментальных и генеративных моделей
FC-2.3	Исследует и создает мульти-модальные большие языковые модели (LLM)
	<p>Знать: архитектуры мульти-модальных больших языковых моделей</p> <p>Уметь: исследовать и создавать мульти-модальные LLM</p> <p>Владеть: созданием мульти-модальных языковых моделей</p>

2. Структура и содержание дисциплины

2.1 Распределение трудоёмкости дисциплины по видам работ

Общая трудоёмкость дисциплины составляет 4 зач. ед. (144 часов), их распределение по видам работ представлено в таблице

Вид учебной работы	Всего часов	Семестры (часы)					
		6					
Контактная работа, в том числе:	84,3	84,3					
Аудиторные занятия (всего):	86	86					
Занятия лекционного типа	32	32					
Лабораторные занятия	48	48					
Занятия семинарского типа (семинары, практические занятия)							
Иная контактная работа:	4,3	4,3					
Контроль самостоятельной работы (КСР)	4	4					
Промежуточная аттестация (ИКР)	0,3	0,3					
Самостоятельная работа, в том числе:	24	24					

Курсовая работа							
Проработка учебного (теоретического) материала	10	10					
Выполнение индивидуальных заданий (подготовка сообщений, презентаций)	14	14					
Реферат							
Подготовка к текущему контролю							
Контроль:	35,7	35,7					
Подготовка к экзамену	35,7	35,7					
Общая трудоемкость	час.	144	144				
	в том числе контактная работа	84,3	84,3				
	зач. ед	4	4				

2.2 Структура дисциплины

Распределение видов учебной работы и их трудоемкости по разделам дисциплины.

Разделы (темы) дисциплины, изучаемые в 6 семестре

№	Наименование разделов (тем)	Количество часов				
		Всего	Аудиторная работа			Внеаудиторная работа СРС
			Л	ПЗ	ЛР	
1	2	3	4	5	6	7
1.	Введение в NLP	5	2		2	1
2.	Лингвистические основы NLP	5	2		2	1
3.	Предобработка текста	6	2		2	2
4.	Векторизация текста: TF-IDF, Word2Vec	8	2		4	2
5.	Рекуррентные сети (RNN, LSTM)	8	2		4	2
6.	Механизм внимания и Transformer	8	2		4	2
7.	BERT и трансферное обучение	7	2		4	1
8.	Генеративные модели (GPT, T5)	8	2		4	2
9.	Оценка качества моделей (BLEU, ROUGE)	6	2		2	2
10	Дообучение и адаптация LLM	7	2		4	1
11	RAG (Retrieval-Augmented Generation)	7	2		4	1
12	Мультимодальные NLP-модели	8	2		4	2
13	Этика и bias в NLP	5	2		2	1
14	Промышленное внедрение NLP-моделей	5	2		2	1
15	Оптимизация NLP-моделей (квантование, дистилляция)	6	2		2	2
16	Итоговый обзор и перспективы NLP	5	2		2	1
ИТОГО по разделам дисциплины		104	32		48	24
Контроль самостоятельной работы (КСР)		4				
Промежуточная аттестация (ИКР)		0,3				
Подготовка к текущему контролю		35,7				
Общая трудоемкость по дисциплине		144				

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 NLP	Определение и области применения NLP. Основные задачи: классификация, генерация, перевод. Обзор инструментов (NLTK, spaCy, Hugging Face). История развития: от rule-based к нейросетевым моделям	Т
2.	Лингвистические основы NLP	Уровни лингвистического анализа. Морфология: части речи, морфологические признаки. Синтаксис: деревья зависимостей. Семантика и прагматика	Т, ЛР
3.	Предобработка текста	Токенизация (слов, предложений). Нормализация: стемминг, лемматизация. Очистка текста (стоп-слова, пунктуация). Регулярные выражения для текста	Т, ЛР
4.	Векторизация текста: TF-IDF, Word2Vec	One-Hot Encoding. Bag of Words (BoW). TF-IDF: принцип и применение. Преимущества и ограничения методов	Т, ЛР
5.	Рекуррентные сети (RNN, LSTM)	Принцип работы RNN. Проблемы RNN: исчезающие градиенты. Архитектуры LSTM и GRU. Примеры применения для текста	Т, ЛР
6.	Механизм внимания и Transformer	Ограничения RNN. Принцип self-attention. Архитектура Transformer. Positional Encoding	Т, ЛР
7.	BERT и трансферное обучение	Архитектура BERT. Предобучение: MLM и NSP. Fine-tuning для downstream-задач. Примеры применения	ЛР
8.	Генеративные модели (GPT, T5)	Архитектура GPT. Принцип генерации текста. Отличия GPT от BERT. Примеры применения (ChatGPT)	ЛР
9	Оценка качества моделей (BLEU, ROUGE)	Метрики для классификации (F1, Accuracy). BLEU для машинного перевода. ROUGE для суммаризации. Human evaluation	ЛР
10	Дообучение и адаптация LLM	Методы дообучения. Подбор learning rate. Адаптация к домену. Catastrophic forgetting	ЛР
11	RAG (Retrieval-Augmented Generation)	Принцип Retrieval-Augmented Generation. Векторные хранилища. Retriever vs Reader. Оптимизация latency	ЛР
12	Мультимодальные NLP-модели	Текст + изображения (CLIP). Текст + аудио (Whisper). Архитектуры мультимодальных моделей. Примеры применения	ЛР
13	Этика и bias в NLP	Bias в данных и моделях. Проблема токсичности. Deepfake-тексты. Ответственное использование ИИ	ЛР

№	Наименование раздела (темы)	Содержание раздела (темы)	Форма текущего контроля
1	2	3	4
14	Промышленное внедрение NLP-моделей	Деплой моделей. Оптимизация для production. Мониторинг моделей. MLOps для NLP	Т
15	Оптимизация NLP-моделей (квантование, дистилляция)	Квантование. Дистилляция. Прунинг. Сравнение методов	ЛР
16	Итоговый обзор и перспективы NLP	Основные тренды. Перспективные направления. Карьера в NLP	Т

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.	Лингвистические основы NLP	ЛБ1 Анализ лингвистических особенностей текста: методы лингвистического анализа текста с помощью библиотек spaCy и NLTK	Т, ЛР
2.	Предобработка текста	ЛБ2 Подготовка текстовых данных для анализа: Токенизация текста на слова и предложения Нормализация (стемминг, лемматизация) Очистка от стоп-слов, пунктуации, чисел Создание собственного пайплайна предобработки	Т, ЛР
3.	Векторизация текста: TF-IDF, Word2Vec	ЛБ3 Преобразование текста в числовые представления: Реализация Bag of Words и TF-IDF (sklearn) Обучение модели Word2Vec (gensim) Визуализация эмбедингов (t-SNE) Сравнение методов векторизации	Т, ЛР
4.	Рекуррентные сети (RNN, LSTM)	ЛБ4 Классификация текста с помощью RNN: Построение модели на базе SimpleRNN Реализация LSTM для анализа тональности Анализ градиентов в RNN	Т, ЛР

№	Наименование раздела (темы)	Наименование лабораторных работ	Форма текущего контроля
1	2	3	4
		Сравнение производительности RNN и линейных моделей	
5.	Механизм внимания и Transformer	ЛБ5 Реализация механизма внимания: Кодирование позиций (Positional Encoding) Построение Self-Attention с нуля (PyTorch) Визуализация attention-карт Сравнение с LSTM на задаче перевода	Т, ЛР
6.	BERT и трансферное обучение	ЛБ6 Fine-tuning BERT для классификации: Загрузка предобученной модели (Hugging Face) Адаптация BERT для анализа тональности Анализ влияния learning rate на качество Интерпретация выходов BERT (Attention Heads)	ЛР
7.	Генеративные модели (GPT, T5)	ЛБ7 Генерация текста с помощью GPT-2: Настройка генерации (temperature, top-k sampling) Дообучение GPT-2 на доменных данных Решение задач с T5 (суммаризация, перевод) Оценка когерентности сгенерированного текста ЛБ15 Разработка кастомной архитектуры Transformer для задачи классификации юридических документов	ЛР
8.	Оценка качества моделей (BLEU, ROUGE)	ЛБ8 Сравнение метрик для NLP-задач: Расчет BLEU для машинного перевода Применение ROUGE для суммаризации Human Evaluation: создание анкет для оценки	ЛР
9	Дообучение и адаптация LLM	ЛБ9 Адаптация GPT-3 под специфичную задачу: Подготовка датасета для дообучения P-tuning vs Fine-tuning Оценка влияния размера датасета на качество Catastrophic Forgetting: методы борьбы Модификация архитектуры Transformer для специализированной NLP-задачи ЛБ14 Анализ ограничений автоматических метрик	ЛР
10	RAG (Retrieval-Augmented Generation)	ЛБ10 Построение QA-системы с RAG: Индексация документов (FAISS) Настройка Retriever и Reader Оптимизация latency системы Сравнение с классическим fine-tuning	ЛР

№	Наименование раздела (темы)	Наименование лабораторных работ	Форма текущего контроля
1	2	3	4
11	Мультимодальные NLP-модели	ЛБ11 Анализ изображений с текстовым описанием: Загрузка и обработка данных (COCO dataset) Использование CLIP для поиска изображений по тексту Генерация описаний с помощью BLIP Оценка качества мультимодальных моделей	ЛР
12	Этика и bias в NLP	ЛБ12 Выявление bias в языковых моделях: Анализ предвзятости в BERT (HONEST-тест) Детекция токсичности (Perspective API) Методы дебиасинга (Data Augmentation) Разработка этических рекомендаций для моделей	ЛР
13	Оптимизация NLP-моделей (квантование, дистилляция)	ЛБ13 Сжатие BERT для edge-устройств: Квантование модели (8-bit) Применение дистилляции (DistilBERT) Замеры скорости инференса до/после оптимизации Развертывание на мобильном устройстве (ONNX)	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)

Не предусмотрены

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 кафедрой информационных технологий, протокол №1 от 30.08.2019
2	Решение задач	Методические указания по организации самостоятельной работы студентов, утвержденные кафедрой информационных технологий, протокол №1 от 30.08.2019

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
	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 «название дисциплины».

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

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 экзамене;

– 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 NLP	DL-1.1; DL-1.3; DL-1.5;	<i>Тест</i>	<i>Вопросы к экзамену 1-2</i>

2	Лингвистические основы NLP	DL-2.1; DL-2.2; DL-4.1; DL-4.2;	Лабораторная работа №1	Вопросы к экзамену 3, проектное задание
3	Предобработка текста	DL-2.1; DL-2.2; DL-4.1; DL-4.2;	Лабораторная работа №2	Вопросы к экзамену 4, проектное задание
4	Векторизация текста: TF-IDF, Word2Vec	DL-4.4; LLM-1.1; FC-2.3	Лабораторная работа №3	Вопросы к экзамену 5, проектное задание
5	Рекуррентные сети (RNN, LSTM)	DL-4.4; LLM-1.1; FC-2.3	Лабораторная работа №4	Вопросы к экзамену 6-7, проектное задание
6	Механизм внимания и Transformer	DL-4.4; LLM-1.1; FC-2.3	Лабораторная работа №5	Вопросы к экзамену 8-10, проектное задание
7	BERT и трансферное обучение	DL-2.1; DL-2.2; DL-4.1; DL-4.2; DL-4.3; DL-4.4;	Лабораторная работа №6	Вопросы к экзамену 11,15, проектное задание
8	Генеративные модели (GPT, T5)	DL-2.1; DL-2.2; DL-4.1; DL-4.2; DL-4.3; DL-4.4;	Лабораторная работа №7, №15	Вопросы к экзамену 12, проектное задание
9	Оценка качества моделей (BLEU, ROUGE)	DL-2.1; DL-2.2; DL-4.1; DL-4.2; DL-4.3; DL-4.4;	Лабораторная работа №8	Вопросы к экзамену 13-14, проектное задание
10	Дообучение и адаптация LLM	DL-4.2; DL-4.3; DL-4.4; LLM-1.1; FC-2.3	Лабораторная работа №9, №14	Вопросы к экзамену 16, проектное задание
11	RAG (Retrieval-Augmented Generation)	DL-4.2; DL-4.3; DL-4.4; LLM-1.1; FC-2.3	Лабораторная работа №10	Вопросы к экзамену 17, проектное задание
12	Мультимодальные NLP-модели	DL-4.2; DL-4.3; DL-4.4; LLM-1.1; FC-2.3	Лабораторная работа №11	Вопросы к экзамену 18, проектное задание
13	Этика и bias в NLP	DL-4.2; DL-4.3; DL-4.4; LLM-1.1; FC-2.3	Лабораторная работа №12	Вопросы к экзамену 19-20, проектное задание
14	Промышленное внедрение NLP-моделей	DL-4.2; DL-4.3; DL-4.4; LLM-1.1; FC-2.3	Тест	Вопросы к экзамену 11, 17, проектное задание
15	Оптимизация NLP-моделей (квантование, дистилляция)	DL-2.1; DL-2.2; DL-4.1; DL-4.2; DL-4.3; DL-4.4	Лабораторная работа №13	Вопросы к экзамену 16, проектное задание

16	Итоговый обзор и перспективы NLP	DL-2.1; DL-2.2; DL-4.1; DL-4.2; DL-4.3; DL-4.4	Тест	Вопросы к экзамену 1, 12, 20
----	----------------------------------	--	------	------------------------------

Показатели, критерии и шкала оценки сформированных компетенций

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

DL-1	Способен применять и (или) разрабатывать архитектуры глубоких нейронных сетей.
DL-1.1	Способен объяснять и применять математические основы нейронных сетей, включая расчет градиентов, методы оптимизации и алгоритм обратного распространения ошибки (backpropagation), для эффективного обучения моделей.
	<p>Знать: основные понятия и формулы для расчета градиентов, базовые методы оптимизации (SGD, Adam) и принцип алгоритма обратного распространения ошибки.</p> <p>Уметь: рассчитывать градиенты для простых функций и реализовывать прямое и обратное распространение для однослойной сети на примере задачи бинарной классификации.</p> <p>Владеть: навыками использования фреймворков (PyTorch/TensorFlow) для автоматического дифференцирования и настройки базовых оптимизаторов.</p>
DL-1.3	Способен применять современные архитектуры глубоких сетей для решения различных задач, понимая их внутреннюю структуру и особенности обучения.
	<p>Знать: основные архитектуры для NLP: RNN (LSTM/GRU), CNN для текста, базовые принципы Transformer-энкодера.</p> <p>Уметь: применять предобученные модели (например, BERT, GPT) через Hugging Face Transformers для стандартных задач (классификация, NER).</p> <p>Владеть: методами тонкой настройки (fine-tuning) готовых моделей на специфическом наборе текстовых данных.</p>
DL-1.5	Способен разрабатывать, адаптировать и внедрять генеративные нейронные сети для решения практических задач, включая создание новых архитектур, оптимизацию обучения и промышленное развертывание моделей
	<p>Знать: основные типы генеративных архитектур (GAN, VAE, авторегрессионные модели like GPT) и их принципиальные различия; понимать архитектуру Encoder-Decoder для задач перевода и суммаризации.</p> <p>Уметь: использовать предобученные генеративные модели (например, T5 для суммаризации, GPT для продолжения текста) через API; генерировать тексты с базовыми техниками (top-k, nucleus sampling).</p> <p>Владеть: базовыми навыками тонкой настройки готовых генеративных моделей на специфические данные.</p>

DL-1.8	Способен разрабатывать, адаптировать и внедрять трансформерные архитектуры для решения задач обработки последовательностей, включая создание новых моделей, оптимизацию обучения и промышленное развертывание
	<p>Знать: базовые компоненты архитектуры Transformer (самовнимание, позиционное кодирование, feed-forward слои); основные типы моделей (encoder-only, decoder-only, encoder-decoder).</p> <p>Уметь: применять стандартные Transformer-модели из библиотек (Hugging Face) для задач классификации, генерации и извлечения признаков.</p> <p>Владеть: навыками тонкой настройки Transformer-моделей на downstream-задачах.</p>
<i>DL-2</i>	<i>Способен применять и (или) разрабатывать современные архитектуры генеративных глубоких сетей</i>
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
	<p>Знать: основные известные архитектуры генеративных сетей (GPT, T5, BART) и их предназначение.</p> <p>Уметь: применять предобученные генеративные модели через высокоуровневые API; проводить простое дообучение (fine-tuning) на предоставленных данных.</p> <p>Владеть: базовыми методами дообучения генеративных моделей на наборах данных.</p>
DL-2.2	Имплементирует известные архитектуры генеративных сетей, реализует пайплайны их обучения на датасетах и вывод генеративных моделей в продуктивную среду
	<p>Знать: основные этапы пайплайна обучения нейронной сети: подготовка данных, инициализация, обучение, валидация.</p> <p>Уметь: запускать готовые скрипты для обучения стандартных генеративных моделей на предоставленных датасетах.</p> <p>Владеть: базовыми навыками работы с пайплайнами обучения из популярных фреймворков (например, Hugging Face Trainer).</p>
<i>DL-4</i>	<i>Способен применять и (или) разрабатывать алгоритмы, методы и технологии обработки естественного языка</i>
DL-4.1	Имплементирует известные алгоритмы, архитектуры и модели обработки естественного языка на реальных данных, строит пайплайны обучения моделей и развертывания NLP-сервисов в продуктивной среде
	<p>Знать: основные этапы NLP-пайплайна (токенизация, векторизация, модель, постобработка); стандартные модели для классических задач.</p> <p>Уметь: имплементировать стандартные NLP-модели (например, на основе TF-IDF + классификатор или fine-tuning BERT) на реальных данных, используя готовые библиотеки.</p>

	Владеть: базовыми навыками развертывания простого NLP-сервиса (например, как веб-API с помощью FCask/FastAPI).
DL-4.2	Определяет стек технологий, методов и алгоритмов для построения продуктов с обработкой естественного языка (диалоговые системы, вопросно-ответные системы, рекомендательные системы и т.д.)
	<p>Знать: основные классы моделей для векторизации текста (Bag-of-Words, TF-IDF, Word2Vec, GloVe) и базовые архитектуры нейросетей (RNN, Transformer).</p> <p>Уметь: применять классические методы (TF-IDF) и простые нейросетевые модели (FastText) для извлечения признаков из текста.</p> <p>Владеть: навыками поиска и применения подходящей предобученной модели для базовой задачи векторизации.</p>
DL-4.3	Имплементирует известные алгоритмы, архитектуры и модели обработки естественного языка на реальных данных, строит пайплайны обучения моделей и развертывания NLP-сервисов в продуктивной среде
	<p>Знать: классические алгоритмы тематического моделирования (LSA, LDA) и базовые подходы к классификации текста.</p> <p>Уметь: применять алгоритмы LDA для тематического моделирования; обучать простые классификаторы (например, на Logistic Regression) на векторизованных TF-IDF признаках.</p> <p>Владеть: навыками применения классических ML-алгоритмов для анализа текстовых данных.</p>
DL-4.4	Разрабатывает новые алгоритмы и библиотеки обработки естественного языка, новые архитектуры глубоких нейронных сетей и методы их обучения для задач анализа текста
	<p>Знать: базовые архитектуры RNN и Transformer, применяемые для стандартных downstream-задач.</p> <p>Уметь: применять предобученные RNN и Transformer модели без модификации их архитектуры для решения задач.</p> <p>Владеть: навыками использования стандартных нейросетевых инструментов для решения задач анализа текста.</p>
LLM-1	<i>Способен применять и (или) разрабатывать генеративные модели и БЯМ</i>
LLM-1.1	Знает архитектуры генеративных моделей.
	<p>Знать: названия и основные принципы ключевых архитектур генеративных моделей (GPT, VAE, GAN, Diffusion Models).</p> <p>Уметь: различать архитектуры генеративных моделей по их описанию или схеме.</p> <p>Владеть: базовой терминологией в области генеративных моделей.</p>
FC-2	<i>Способен проводить фронтальные исследования в области фундаментальных и генеративных моделей</i>
FC-2.3	Исследует и создает мульти-модальные большие языковые модели (LLM)

	<p>Знать: базовые принципы построения мультимодальных LLM (например, LLaVA, BLIP), способы объединения текстовой и визуальной модальностей.</p> <p>Уметь: использовать существующие мультимодальные LLM через API или готовые реализации для решения простых задач (например, описание изображения).</p> <p>Владеть: пониманием концепции мультимодальности и навыками применения готовых мультимодальных моделей.</p>
--	--

Соответствие **базовому уровню** освоения компетенций планируемым результатам обучения и критериям их оценивания (оценка: **хорошо /зачтено**):

<i>DL-1</i>	<i>Способен применять и (или) разрабатывать архитектуры глубоких нейронных сетей.</i>
DL-1.1	Способен объяснять и применять математические основы нейронных сетей, включая расчет градиентов, методы оптимизации и алгоритм обратного распространения ошибки (backpropagation), для эффективного обучения моделей.
	<p>Знать: детали расчета градиентов для рекуррентных и сверточных архитектур, применяемых в NLP; особенности работы алгоритма backpropagation through time (BPTT); методы борьбы с исчезающими градиентами.</p> <p>Уметь: модифицировать функции потерь; настраивать гиперпараметры оптимизатора (learning rate, вес decay) для улучшения сходимости моделей на текстовых данных.</p> <p>Владеть: методами отладки и контроля процесса обучения нейронных сетей для задач NLP (анализ кривых обучения, использование градиентного клиппинга).</p>
DL-1.3	Способен применять современные архитектуры глубоких сетей для решения различных задач, понимая их внутреннюю структуру и особенности обучения.
	<p>Знать: сильные и слабые стороны различных архитектур (RNN vs CNN vs Transformer) для разных типов NLP-задач; особенности их обучения и тонкой настройки.</p> <p>Уметь: модифицировать стандартные архитектуры (например, добавлять механизмы внимания к RNN, использовать различные конфигурации слоев в Transformer); проводить transfer learning для доменно-специфичных данных.</p> <p>Владеть: навыками выбора и адаптации архитектуры глубокой сети под требования конкретной задачи и имеющихся вычислительных ресурсов.</p>
DL-1.5	Способен разрабатывать, адаптировать и внедрять генеративные нейронные сети для решения практических задач, включая создание новых архитектур, оптимизацию обучения и промышленное развертывание моделей
	Знать: детали архитектур (например, механизм внимания в Seq2Seq), техники борьбы с распространенными проблемами (repetition, exposure bias), методы контролируемой генерации.

	<p>Уметь: дообучать существующие генеративные модели на специфических данных; настраивать параметры генерации; проводить анализ ошибок модели и применять методы регуляризации и аугментации данных.</p> <p>Владеть: методами адаптации и тонкой настройки генеративных моделей под требования конкретного продукта.</p>
DL-1.8	Способен разрабатывать, адаптировать и внедрять трансформерные архитектуры для решения задач обработки последовательностей, включая создание новых моделей, оптимизацию обучения и промышленное развертывание
	<p>Знать: детали механизма внимания (multi-head, scaled dot-product), способы позиционного кодирования, особенности тонкой настройки (fine-tuning) трансформеров.</p> <p>Уметь: адаптировать трансформерные архитектуры для специфических задач (например, менять pooling-стратегии, добавлять task-specific heads); использовать эффективные варианты внимания (например, Longformer) для длинных текстов.</p> <p>Владеть: методами эффективной тонкой настройки трансформеров на доменных данных.</p>
DL-2	<i>Способен применять и (или) разрабатывать современные архитектуры генеративных глубоких сетей</i>
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
	<p>Знать: особенности дообучения генеративных моделей (fine-tuning), техники контролируемой генерации (prompt engineering, guidance scale).</p> <p>Уметь: проводить дообучение существующих моделей на специфических данных; настраивать параметры генерации (temperature, top-k, top-p); комбинировать несколько моделей в простой пайплайн.</p> <p>Владеть: методами адаптации генеративных моделей под конкретный домен и задачу.</p>
DL-2.2	Имплементирует известные архитектуры генеративных сетей, реализует пайплайны их обучения на датасетах и вывод генеративных моделей в продуктивную среду
	<p>Знать: детали реализации архитектур генеративных сетей (на уровне кода); принципы построения отказоустойчивых пайплайнов обучения и инференса.</p> <p>Уметь: имплементировать и модифицировать архитектуры генеративных сетей; создавать воспроизводимые пайплайны обучения с логированием и мониторингом.</p> <p>Владеть: навыками разработки и поддержки полного пайплайна обучения генеративной модели, от данных до артефакта модели.</p>
DL-4	<i>Способен применять и (или) разрабатывать алгоритмы, методы и технологии обработки естественного языка</i>

DL-4.1	<p>Имплементирует известные алгоритмы, архитектуры и модели обработки естественного языка на реальных данных, строит пайплайны обучения моделей и развертывания NLP-сервисов в продуктивной среде</p>
	<p>Знать: современные NLP-архитектуры (Transformer, BERT) и методы их применения; основы MLOps для NLP.</p> <p>Уметь: строить полные пайплайны для обучения и оценки моделей на реальных данных; развертывать модели в контейнеризированной среде (Docker); настраивать базовый мониторинг.</p> <p>Владеть: методами построения воспроизводимых пайплайнов обучения и развертывания NLP-моделей в продакшн-подобной среде.</p>
DL-4.2	<p>Определяет стек технологий, методов и алгоритмов для построения продуктов с обработкой естественного языка (диалоговые системы, вопросно-ответные системы, рекомендательные системы и т.д.)</p>
	<p>Знать: сильные и слабые стороны различных методов векторизации и архитектур; типовые архитектурные решения для разных классов NLP-продуктов (QA, диалоги).</p> <p>Уметь: обоснованно выбирать стек технологий (модели, фреймворки, инфраструктуру) для конкретной продуктовой задачи, учитывая ограничения по качеству и производительности.</p> <p>Владеть: методами сравнительного анализа и выбора оптимального алгоритмического решения для построения NLP-продукта.</p>
DL-4.3	<p>Имплементирует известные алгоритмы, архитектуры и модели обработки естественного языка на реальных данных, строит пайплайны обучения моделей и развертывания NLP-сервисов в продуктивной среде</p>
	<p>Знать: архитектуры рекуррентных сетей (LSTM) и трансформеров, принципы их работы и области применения в downstream-задачах.</p> <p>Уметь: обучать модели классификации и др. задач с использованием плотных векторных представлений от RNN/Transformer; проводить тонкую настройку предобученных энкодеров (BERT).</p> <p>Владеть: методами применения современных нейросетевых архитектур для решения практических NLP-задач.</p>
DL-4.4	<p>Разрабатывает новые алгоритмы и библиотеки обработки естественного языка, новые архитектуры глубоких нейронных сетей и методы их обучения для задач анализа текста</p>
	<p>Знать: детали архитектур и возможности их модификации (изменение механизмов внимания, добавление новых слоев).</p> <p>Уметь: модифицировать существующие архитектуры (например, добавлять кастомные слои или механизмы внимания) для улучшения качества решения конкретных задач.</p> <p>Владеть: методами адаптации и улучшения нейросетевых архитектур под специфические требования.</p>
LLM-1	<i>Способен применять и (или) разрабатывать генеративные модели и БЯМ</i>
LLM-1.1	Знает архитектуры генеративных моделей.

	<p>Знать: ключевые различия, преимущества и недостатки основных архитектур (GPT vs T5); понимать эволюцию архитектур (от RNN до Transformer).</p> <p>Уметь: сравнивать архитектуры и обоснованно выбирать подходящую для решения конкретной прикладной задачи (например, GPT для генерации, T5 для перефразирования).</p> <p>Владеть: навыками сравнительного анализа генеративных архитектур.</p>
FC-2	<i>Способен проводить фронтальные исследования в области фундаментальных и генеративных моделей</i>
FC-2.3	Исследует и создает мульти-модальные большие языковые модели (LLM)
	<p>Знать: архитектурные подходы к созданию мультимодальных моделей (alignment модули, cross-attention), методы их обучения (контрастивные, generative).</p> <p>Уметь: проводить дообучение (fine-tuning) существующих мультимодальных LLM на специфических наборах данных; оценивать качество их работы.</p> <p>Владеть: методами адаптации мультимодальных LLM под новые домены и задачи.</p>

Соответствие **продвинутому уровню** освоения компетенций планируемым результатам обучения и критериям их оценивания (оценка: **отлично /зачтено**):

DL-1	<i>Способен применять и (или) разрабатывать архитектуры глубоких нейронных сетей.</i>
DL-1.1	Способен объяснять и применять математические основы нейронных сетей, включая расчет градиентов, методы оптимизации и алгоритм обратного распространения ошибки (backpropagation), для эффективного обучения моделей.
	<p>Знать: современные методы оптимизации второго порядка, принципы работы адаптивных методов; математические основы техник ускоренного обучения (например, Lookahead, LAMB).</p> <p>Уметь: реализовывать кастомные функции потерь для специфических NLP-задач (контрастивное обучение, метрическое обучение); диагностировать и устранять сложные проблемы обучения, связанные с динамикой градиентов.</p> <p>Владеть: навыками тонкой настройки процесса оптимизации для достижения state-of-the-art результатов на конкретных датасетах и задачах.</p>
DL-1.3	Способен применять современные архитектуры глубоких сетей для решения различных задач, понимая их внутреннюю структуру и особенности обучения.
	<p>Знать: передовые и гибридные архитектуры (например, Transformer-XL, Performer, архитектуры с кэшированием), принципы sparse models, методы дистилляции знаний.</p> <p>Уметь: разрабатывать и имплементировать кастомные архитектуры для специфических задач (например, для совместной обработки текста и структурированных данных); оптимизировать совместное обучение нескольких задач (multi-task learning).</p>

	<p>Владеть: методами создания и обучения эффективных пайплайнов для сложных NLP-приложений, включая техники сжатия и ускорения моделей.</p>
DL-1.5	<p>Способен разрабатывать, адаптировать и внедрять генеративные нейронные сети для решения практических задач, включая создание новых архитектур, оптимизацию обучения и промышленное развертывание моделей</p>
	<p>Знать: современные архитектуры (T5, BART, диффузионные модели для текста), продвинутое обучение (с подкреплением, контрастивное обучение), методы эффективного обучения больших моделей (LoRA, QLoRA).</p> <p>Уметь: разрабатывать новые архитектуры или гибридные подходы; оптимизировать процесс обучения для генеративных моделей; реализовывать сложные пайплайны генерации, включающие несколько моделей.</p> <p>Владеть: полным циклом разработки и промышленного развертывания высоконагруженных генеративных NLP-сервисов.</p>
DL-1.8	<p>Способен разрабатывать, адаптировать и внедрять трансформерные архитектуры для решения задач обработки последовательностей, включая создание новых моделей, оптимизацию обучения и промышленное развертывание</p>
	<p>Знать: современные вариации трансформеров (Sparse, Linformer, Performer), методы ускорения инференса, техники дистилляции и квантования трансформерных моделей.</p> <p>Уметь: модифицировать архитектуру трансформера для исследовательских целей; разрабатывать и внедрять оптимизированные версии моделей для промышленного использования.</p> <p>Владеть: навыками промышленного развертывания и обслуживания больших трансформерных моделей, включая оптимизацию для GPU/CPU и управление версиями.</p>
DL-2	<p><i>Способен применять и (или) разрабатывать современные архитектуры генеративных глубоких сетей</i></p>
DL-2.1	<p>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</p>
	<p>Знать: продвинутое обучение (P-tuning, LoRA, QLoRA), техники few-shot/zero-shot обучения, методы RLHF (Reinforcement Learning from Human Feedback).</p> <p>Уметь: разрабатывать сложные гибридные пайплайны генерации; модифицировать архитектуры моделей; применять reinforcement learning для оптимизации процесса генерации под заданную метрику.</p> <p>Владеть: полным циклом адаптации, оптимизации и развертывания кастомных генеративных решений.</p>
DL-2.2	<p>Имплементирует известные архитектуры генеративных сетей, реализует пайплайны их обучения на датасетах и вывод генеративных моделей в продуктивную среду</p>

	<p>Знать: передовые практики MLOps для генеративных моделей, методы оптимизации инференса (ONNX, TensorRT), принципы построения high-load сервисов.</p> <p>Уметь: разрабатывать кастомные высокооптимизированные реализации архитектур; создавать масштабируемые и отказоустойчивые сервисы для генерации; реализовывать сложные системы А/В-тестирования.</p> <p>Владеть: методами промышленного развертывания и обслуживания генеративных моделей в высоконагруженной продуктивной среде.</p>
DL-4	<i>Способен применять и (или) разрабатывать алгоритмы, методы и технологии обработки естественного языка</i>
DL-4.1	Имплементирует известные алгоритмы, архитектуры и модели обработки естественного языка на реальных данных, строит пайплайны обучения моделей и развертывания NLP-сервисов в продуктивной среде
	<p>Знать: передовые методы обработки данных, аугментации, активного обучения; принципы построения распределенных и отказоустойчивых сервисов.</p> <p>Уметь: разрабатывать кастомные инструменты для предобработки и разметки текстов; создавать высоконагруженные, масштабируемые NLP-сервисы с кешированием и балансировкой нагрузки.</p> <p>Владеть: навыками построения и управления комплексной MLOps-инфраструктурой для жизненного цикла NLP-моделей.</p>
DL-4.2	Определяет стек технологий, методов и алгоритмов для построения продуктов с обработкой естественного языка (диалоговые системы, вопросно-ответные системы, рекомендательные системы и т.д.)
	<p>Знать: state-of-the-art модели и архитектуры, включая крупные языковые модели (LLMs) и мультимодальные подходы; экономику и ограничения их применения.</p> <p>Уметь: проектировать сложные, гибридные системы, комбинирующие различные подходы (нейросетевые, графовые, символьные); закладывать в архитектуру системы возможности для дальнейшего масштабирования и улучшения.</p> <p>Владеть: компетенцией по архитектурному проектированию комплексных NLP-продуктов.</p>
DL-4.3	Имплементирует известные алгоритмы, архитектуры и модели обработки естественного языка на реальных данных, строит пайплайны обучения моделей и развертывания NLP-сервисов в продуктивной среде
	<p>Знать: внутреннее устройство и особенности обучения больших языковых моделей (LLMs), техники адаптации контекста (in-context learning), продвинутые методы генерации.</p> <p>Уметь: адаптировать и дообучать большие языковые модели для специфических downstream-задач, требующих глубокого понимания контекста и генерации связного текста.</p> <p>Владеть: продвинутыми техниками работы с LLM, включая prompt engineering, fine-tuning и оценку качества генерации.</p>

DL-4.4	Разрабатывает новые алгоритмы и библиотеки обработки естественного языка, новые архитектуры глубоких нейронных сетей и методы их обучения для задач анализа текста
	<p>Знать: фундаментальные принципы построения нейросетевых архитектур, последние научные достижения в области NLP.</p> <p>Уметь: проводить исследования и предлагать принципиально новые архитектуры или алгоритмы обработки текста; разрабатывать и публиковать opensource-библиотеки.</p> <p>Владеть: методологией проведения фронтальных исследований и создания прототипов новых алгоритмов и моделей.</p>
LLM-1	<i>Способен применять и (или) разрабатывать генеративные модели и БЯМ</i>
LLM-1.1	Знает архитектуры генеративных моделей.
	<p>Знать: тонкие архитектурные детали и модификации state-of-the-art моделей (например, sparse attention, mixture-of-experts), а также направления их развития.</p> <p>Уметь: критически оценивать и предлагать модификации существующих архитектур для решения новых исследовательских или прикладных проблем.</p> <p>Владеть: глубоким экспертным знанием в области архитектур генеративных моделей.</p>
FC-2	<i>Способен проводить фронтальные исследования в области фундаментальных и генеративных моделей</i>
FC-2.3	Исследует и создает мульти-модальные большие языковые модели (LLM)
	<p>Знать: современные research направления в области мультимодальных LLM, вызовы и ограничения; передовые методы обучения и выравнивания модальностей.</p> <p>Уметь: разрабатывать и имплементировать новые архитектурные решения для мультимодальных LLM; проводить полный цикл исследований от идеи до экспериментов и публикации.</p> <p>Владеть: компетенцией по проведению фронтальных исследований и созданию прототипов новых мультимодальных архитектур.</p>

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

1. Лингвистический анализ текста

Кейс: Разработка системы автоматической разметки медицинских документов

Постановка задачи:

Имеется набор медицинских заключений с неструктурированным текстом

Требуется:

Выделить именованные сущности (болезни, препараты, процедуры)

Определить семантические связи между ними

Построить граф зависимостей для сложных диагнозов

Инструменты: spaCy, Prodigy, UMLS-онтология

2. Предобработка текста

Кейс: Подготовка пользовательских отзывов для анализа

Постановка задачи:

Собраны 50 000 отзывов с орфографическими ошибками и сленгом

Требуется:

Создать конвейер очистки (нормализация, исправление ошибок)

Разработать словарь для замены сленговых выражений

Обеспечить обработку 1000 документов/сек

Инструменты: SymSpell, FastText, регулярные выражения

3. Векторизация текста

Кейс: Построение рекомендательной системы для новостного портала

Постановка задачи:

Имеется архив 1 млн. новостных статей

Требуется:

Реализовать 3 метода векторизации (BoW, TF-IDF, Doc2Vec)

Сравнить качество рекомендаций по метрике precision@k

Оптимизировать скорость формирования рекомендаций

Инструменты: Gensim, Annoy, Faiss

4. Рекуррентные сети

Кейс: Детекция аномальных транзакций по описаниям

Постановка задачи:

Входные данные: текстовые описания банковских операций

Требуется:

Построить LSTM-модель для классификации аномалий

Обеспечить интерпретируемость предсказаний

Достичь F1-score ≥ 0.92 на несбалансированных данных

Инструменты: PyTorch, Captum, imbalanced-learn

5. Трансформеры

Кейс: Многоязычный чат-бот для авиакомпании

Постановка задачи:

Необходимо обрабатывать запросы на 5 языках

Требуется:

Адаптировать Transformer-модель для параллельной обработки языков

Реализовать механизм переключения языка

Обеспечить время ответа < 500 мс

Инструменты: Hugging Face Transformers, ONNX Runtime

6. BERT и трансферное обучение

Кейс: Автоматизация обработки юридических документов

Постановка задачи:

Архив договоров (10 000+ страниц)

Требуется:

Дообучить BERT для выделения ключевых условий

Классифицировать документы по 20 категориям

Обеспечить точность $\geq 88\%$ на редких категориях

Инструменты: BERTopic, Label Studio

7. Генеративные модели

Кейс: Создание контент-плана для соцсетей

Постановка задачи:

Имеется база 100 000 постов конкурентов

Требуется:

Настроить GPT-3 для генерации креативных заголовков

Реализовать контроль стилистики (формальный/неформальный)

Интегрировать с CMS через API

Инструменты: OpenAI API, StyleGAN-2

8. Оценка качества

Кейс: Валидация системы автоматического перевода

Постановка задачи:

Разработана модель перевода технической документации

Требуется:

Сравнить 3 метрики оценки (BLEU, TER, METEOR)

Провести human evaluation с участием 10 экспертов

Выявить системные ошибки перевода

Инструменты: SacreBLEU, Amazon Mechanical Turk

9. Дообучение LLM

Кейс: Адаптация модели для финансового сектора

Постановка задачи:

Необходимо анализировать отчеты компаний

Требуется:

Собрать корпус из 50 000 финансовых документов

Применить P-tuning для адаптации GPT-3

Оценить прирост качества на доменных запросах

Инструменты: LoRA, NVIDIA NeMo

10. RAG

Кейс: Виртуальный ассистент для университета

Постановка задачи:

Имеется база нормативных документов (5000+ страниц)

Требуется:

Реализовать RAG-систему для ответов на вопросы

Интегрировать поиск по PDF-файлам

Обеспечить точность ответов $\geq 85\%$

Инструменты: Haystack, Milvus, Unstructured.io

11. Мультимодальные модели

Кейс: Система контроля качества на производстве

Постановка задачи:

Входные данные: фото изделий + текстовые отчеты

Требуется:

Настроить CLIP для классификации дефектов

Генерировать текстовые описания проблем

Интегрировать с промышленным IoT-оборудованием

Инструменты: OpenCV, NVIDIA Triton

12. Этика в NLP

Кейс: Аудит HR-системы на bias

Постановка задачи:
Имеется модель для анализа резюме
Требуется:
Выявить дискриминацию по полу/возрасту
Перебалансировать обучающие данные
Разработать этические рекомендации
Инструменты: AIF360, Fairlearn

13. Оптимизация моделей

Кейс: Развертывание NLP-модели на смартфонах
Постановка задачи:
Требуется адаптировать BERT для мобильных устройств
Условия:
Размер модели < 50 МБ
Время инференса < 100 мс
Потери точности $\leq 3\%$
Инструменты: TensorFCow Lite, Qualcomm AI Engine

14. Модификация архитектуры Transformer для специализированной NLP-задачи

Кейс: Разработка доменно-специфичной NLP-модели для юридической аналитики
Постановка задачи:
Адаптировать архитектуру Transformer для классификации юридических документов (договоры, иски, нормативные акты) с сохранением точности на длинных текстах.

Условия:
Обработка документов до 10k токенов (в 20 раз больше стандартного BERT)
Поддержка юридической терминологии ($\geq 95\%$ распознавания спец. терминов)
Точность классификации $\geq 85\%$ (F1-score)
Совместимость с ONNX Runtime для промышленного внедрения
Инструменты:
Библиотеки: Hugging Face Transformers, PyTorch, ONNX
Оптимизация: смешанная точность (FP16), прунинг голов внимания
Датасет: CUAD (25k+ юридических документов)

15. Разработка кастомной архитектуры Transformer для задачи классификации юридических документов

Кейс: Развертывание NLP-модели для обработки клинических записей в edge-устройствах

Постановка задачи:
Модифицировать архитектуру Transformer для извлечения медицинских терминов из электронных карт пациента с учетом:
Ограничений мобильных устройств (CPU/GPU)
Конфиденциальности данных (оффлайн-обработка)
Условия:
Размер модели ≤ 30 МБ
Время обработки 1 записи (300-500 токенов) ≤ 50 мс
Точность NER (F1) $\geq 90\%$ для медицинских терминов
Поддержка динамического словаря терминов
Инструменты:
Фреймворки: TensorFCow Lite, ONNX Runtime
Модели: DistilBioBERT (квантованный INT8)
Техники: LoRA-адаптация, knowledge distillation
Датасет: NCBI Disease Corpus

Пример лабораторной работы

Лабораторная работа №2: Предобработка текста

Тема: Подготовка пользовательских отзывов для анализа

Цель работы:

Освоить методы предобработки текстовых данных, включая нормализацию, исправление орфографических ошибок и замену сленга, для последующего анализа с помощью NLP-моделей.

Задачи:

Реализовать конвейер предобработки текста на Python.

Применить алгоритмы исправления орфографии (SymSpell).

Создать и использовать словарь для замены сленговых выражений.

Оптимизировать скорость обработки до 1000 документов/сек.

Ожидаемые результаты:

Готовый Python-скрипт для предобработки текста.

Очищенные от ошибок и сленга тексты отзывов.

Отчет с анализом эффективности методов (точность исправления ошибок, скорость работы).

Ход работы

1. Подготовка данных

Загрузить датасет с отзывами (файл reviews.csv).

Провести выборочный анализ данных для выявления типичных ошибок и сленга.

2. Реализация конвейера предобработки

Этапы:

Токенизация: Разбить текст на слова.

```
from nltk.tokenize import word_tokenize
text = "This is a sample text with slangs like 'gr8' and errors like 'recieve'."
tokens = word_tokenize(text)
```

Нормализация:

Привести текст к нижнему регистру.

Удалить спецсимволы и цифры с помощью регулярных выражений.

```
import re
normalized_text = re.sub(r'^[a-zA-Z\s]', "", text.lower())
```

Исправление орфографии (SymSpell):

```
from symspellpy import SymSpell
sym_spell = SymSpell()
sym_spell.load_dictionary('frequency_dictionary_en_82_765.txt', term_index=0, count_index=1)
corrected_words = [sym_spell.lookup(word, Verbosity.CLOSEST)[0].term for word in tokens]
```

Замена сленга:

Создать словарь сленга (slang_dict = {'gr8': 'great', 'btw': 'by the way'}).

Заменить слова в тексте.

```
def replace_slang(tokens, slang_dict):
```

```
return [slang_dict.get(word, word) for word in tokens]
```

3. Оптимизация скорости

Использовать multiprocessing или pandas.apply() для параллельной обработки.
Замерить время выполнения для 1000 документов.

4. Анализ результатов

Сравнить текст до и после обработки.
Оценить точность исправлений на тестовой выборке.

Требования к отчету

1. Титульный лист

Название работы, ФИО студента, группа, дата.

2. Введение

Цель и задачи работы.
Описание датасета (объем, типы ошибок).

3. Методы

Описание алгоритмов:
Токенизация.
Исправление ошибок (SymSpell).
Замена сленга.

4. Реализация

Код на Python с комментариями.
Примеры до/после обработки.

5. Результаты

Таблица с метриками.

6. Выводы

Проблемы, возникшие при обработке.
Предложения по улучшению.

7. Приложения

Исходный код.
Словарь сленга.

Критерии оценки

Отлично (5): Полная реализация конвейера, скорость >1000 док/сек, детальный анализ ошибок.

Хорошо (4): Реализованы основные этапы, скорость 500–1000 док/сек, отсутствует анализ точности.

Удовлетворительно (3): Конвейер работает, но с ошибками, скорость <500 док/сек.

Неудовлетворительно (2): Критические ошибки в коде или отсутствие ключевых этапов.

Рекомендуемые инструменты

Язык: Python 3.8+.

Библиотеки:

nlk, symspellpy, pandas, re.

Данные: Пример датасета – Yelp Reviews Dataset

Тестовые вопросы по дисциплине

1. Базовые концепции NLP

Какой из методов не относится к предобработке текста?

- a) Токенизация
- b) Лемматизация
- c) Квантование
- d) Удаление стоп-слов

Какая метрика не используется для оценки качества машинного перевода?

- a) BLEU
- b) ROUGE
- c) Accuracy
- d) METEOR

2. Векторизация текста

Какой алгоритм не является методом word embedding?

- a) Word2Vec
- b) TF-IDF
- c) GloVe
- d) FastText

Какой параметр в TF-IDF уменьшает вес часто встречающихся слов?

- a) Term Frequency (TF)
- b) Inverse Document Frequency (IDF)
- c) Нормализация L2
- d) Косинусное сходство

3. Нейросетевые модели

Какая архитектура не является рекуррентной?

- a) LSTM
- b) GRU
- c) Transformer
- d) SimpleRNN

Какой механизм позволяет Transformer учитывать контекст без рекуррентных связей?

- a) Backpropagation
- b) Self-Attention
- c) Dropout
- d) Batch Normalization

4. Трансформеры и BERT

Какой тип обучения используется в BERT?

- a) Только supervised
- b) Только unsupervised
- c) Self-supervised (MLM + NSP)
- d) Reinforcement learning

Какой компонент не входит в архитектуру Transformer?

- a) Positional Encoding

- b) Self-Attention
- c) Сверточные слои
- d) Feed-Forward Network

5. Генеративные модели

Какой параметр GPT-3 влияет на "креативность" генерации?

- a) Learning Rate
- b) Temperature
- c) Batch Size
- d) Dropout Rate

Какая задача не решается с помощью T5?

- a) Суммаризация
- b) Машинный перевод
- c) Классификация изображений
- d) Ответы на вопросы

6. Оценка и оптимизация

Какая метрика лучше подходит для оценки суммаризации?

- a) BLEU
- b) ROUGE
- c) Precision
- d) F1-score

Какой метод не используется для сжатия NLP-моделей?

- a) Квантование
- b) Дистилляция
- c) Прунинг
- d) Аугментация

7. Прикладные аспекты

Какой инструмент не входит в экосистему Hugging Face?

- a) Transformers
- b) Datasets
- c) Tokenizers
- d) TensorBoard

Какой подход используется в RAG для поиска релевантных документов?

- a) Полносвязные нейросети
- b) Векторный поиск (semantic search)
- c) Деревья решений
- d) Метод опорных векторов (SVM)

Какая проблема не связана с bias в NLP?

- a) Генерация токсичного контента
- b) Дискриминация по полу в предсказаниях
- c) Высокая perplexity модели
- d) Стереотипные ассоциации

Ключи к тестовым вопросам

- c) Квантование
- c) Assurasy

- b) TF-IDF
- b) Inverse Document Frequency (IDF)
- c) Transformer
- b) Self-Attention
- c) Self-supervised (MLM + NSP)
- c) Сверточные слои
- b) Temperature
- c) Классификация изображений
- b) ROUGE
- d) Аугментация
- d) TensorBoard
- b) Векторный поиск (semantic search)
- c) Высокая perplexity модели

Время выполнения: 20-30 минут.

Проверяемые компетенции комплексом практических заданий: DL-1.1; DL-1.3; DL-1.5; DL-1.8; DL-2.1; DL-2.2; DL-4.1; DL-4.2; DL-4.3; DL-4.4; LLM-1.1; FC-2.3

Зачетно-экзаменационные материалы для промежуточной аттестации (экзамен)

Вопросы для подготовки к экзамену

1. Определение NLP. Основные задачи и приложения.
2. Уровни лингвистического анализа (морфология, синтаксис, семантика, прагматика).
3. Основные отличия между стеммингом и лемматизацией. Примеры.
4. Принцип работы TF-IDF. Примеры практических задач.
5. Понятие word embeddings. Особенности Word2Vec, GloVe и FastText.
6. Основные принципы архитектуры рекуррентных нейросетей (RNN).

Ограничения.

7. Решение проблемы исчезающих градиентов с помощью LSTM.
8. Принцип работы механизма внимания (attention) в Seq2Seq-моделях.
9. Отличия self-attention от обычного attention.
10. Принципы архитектуры Transformer. Перечень и характеристики компонентов.
11. Принципы работы BERT. Практическое применение.
12. Основные отличия BERT и GPT.
13. Метрики для оценки качества машинного перевода.
14. Способы оценки качества суммаризации текста.
15. Понятие fine-tuning. Примеры downstream-задач.
16. Методы дообучения LLM.
17. Принцип RAG (Retrieval-Augmented Generation).
18. Архитектуры мультимодальных моделей.
19. Понятие bias в NLP. Способы его уменьшения.
20. Этические проблемы использования больших языковых моделей.

Перечень компетенций (части компетенции), проверяемых оценочным средством
DL-1.1; DL-1.3; DL-1.5; DL-1.8; DL-2.1; DL-2.2; DL-4.1; DL-4.2; DL-4.3; DL-4.4; LLM-1.1; FC-2.3

Типовое проектное задание я к экзамену

Проектное задание: Разработка и внедрение NLP-пайплайна для анализа тональности отзывов

Цель:

Создать end-to-end решение для классификации тональности текстовых отзывов (позитивный/негативный/нейтральный) с развертыванием в виде веб-API.

Этапы выполнения:

1. Постановка задачи

Выбрать датасет (например, IMDB Reviews).

Определить метрики качества: Accuracy, F1-score.

2. Предобработка данных

Очистка текста (удаление HTML-тегов, стоп-слов, пунктуации).

Лемматизация/стемминг (spaCy, NLTK).

Векторизация (TF-IDF, Word2Vec, или BERT-токенизация).

3. Выбор и обучение модели

Варианты моделей:

Классические: Logistic Regression + TF-IDF.

Нейросетевые: LSTM, Transformer (BERT).

Обучение и валидация (train/test split, кросс-валидация).

4. Оптимизация

Подбор гиперпараметров (GridSearch/RandomizedSearch).

Квантование модели (для уменьшения размера).

5. Развертывание (Deployment)

Упаковка модели в FastAPI-приложение.

Создание Docker-контейнера.

Тестирование API (например, через Postman).

6. Документирование и отчет

Описание каждого этапа.

Визуализация результатов (матрица ошибок, графики).

Ссылка на GitHub-репозиторий с кодом и инструкцией по запуску.

Перечень компетенций (части компетенции), проверяемых оценочным средством

DL-1.1; DL-1.3; DL-1.5; DL-1.8; DL-2.1; DL-2.2; DL-4.1; DL-4.2; DL-4.3; DL-4.4; LLM-1.1;
FC-2.3

Вопросы на анализ и сравнение

1. Сравните Bag of Words и Word Embeddings: плюсы, минусы, области применения.
2. Почему трансформеры заменили RNN в большинстве NLP-задач?
3. Какие преимущества и недостатки у BERT по сравнению с традиционными word embeddings?
4. Когда стоит использовать rule-based подходы, а когда – нейросетевые?
5. Как можно улучшить качество машинного перевода для низкоресурсных языков?

Перечень компетенций (части компетенции), проверяемых оценочным средством

DL-1.1; DL-1.3; DL-1.5; DL-1.8; DL-2.1; DL-2.2; DL-4.1; DL-4.2; DL-4.3; DL-4.4; LLM-1.1;
FC-2.3

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

Методические рекомендации, определяющие процедуры оценивания на экзамене:

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 экзамен. Студенты обязаны сдать экзамен в соответствии с расписанием и учебным планом.

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

Экзамен 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 экзамена: устно.

Экзаменатору предоставляется право задавать студентам дополнительные вопросы по всей учебной программе дисциплины.

Результат сдачи экзамена заносится преподавателем в экзаменационную ведомость и зачетную книжку.

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

Критерии оценки:

1. Оценка ответов на экзаменационные вопросы (40% итоговой оценки)

Отлично (5)

Полные, развернутые ответы с демонстрацией глубокого понимания темы.

Использование примеров, формул, корректных терминов.

Умение анализировать и сравнивать методы (например, TF-IDF vs. Word2Vec).

% выполнения: 90–100% (допускаются незначительные неточности).

Хорошо (4)

Ответы содержат основные идеи, но без углубленного анализа.

Возможны небольшие ошибки в деталях или формулировках.

% выполнения: 75–89%.

Удовлетворительно (3)

Ответы поверхностные, с существенными пробелами.

Отсутствие примеров или некорректное применение терминов.

% выполнения: 60–74%.

Неудовлетворительно (2)

Отсутствие понимания ключевых концепций.

Грубые ошибки или неспособность ответить на большую часть вопросов.

% выполнения: <60%.

2. Оценка выполнения практических кейсов и лабораторных работ (20% итоговой оценки)

Отлично (5)

Полное выполнение всех этапов кейса с инновационными решениями.

Достижение целевых метрик (например, $F1 > 0.9$).

Четкая документация кода и анализ результатов.

% выполнения: 90–100%.

Хорошо (4)

Выполнены основные задачи, но без дополнительной оптимизации.

Незначительные отклонения от целевых метрик (например, $F1 = 0.85$).

% выполнения: 75–89%.

Удовлетворительно (3)

Решены базовые задачи, но с критическими ошибками.

Низкое качество кода или отсутствие анализа.

% выполнения: 60–74%.

Неудовлетворительно (2)

Невыполнение ключевых этапов.

Код нерабочий или отсутствует.

% выполнения: <60%.

3. Оценка выполнения проектного задания (40%):

Отлично (5)

Полный пайплайн + API, метрики $F1 > 0.9$, оптимизация (квантование).

Хорошо (4)

Рабочий пайплайн без развертывания, $F1 > 0.85$.

Удовлетворительно (3)

Готовая модель, но с ошибками в preprocessing или низким качеством ($F1 < 0.8$).

Неудовлетворительно (2)

Невыполнение ключевых этапов.

Код нерабочий или отсутствует

Итоговая оценка (суммарно)

Оценка	Экзамен (40%)	Практика (20%)	Проект (40%)	Общий %
Отлично (5)	90–100%	90–100%	87–100%	$\geq 85\%$
Хорошо (4)	75–89%	75–89%	67–80%	70–84%
Удовлетворительно (3)	60–74%	60–74%	47–60%	55–69%
Неудовлетворительно (2)	<60%	<60%	<47%	<55%

Для допуска к экзамену необходимо выполнить **все лабораторные работы на минимум "удовлетворительно"**.

"Отлично" требует высоких результатов во всех компонентах (особенно в практических кейсах).

Практические кейсы оцениваются по:

–Корректности кода.

–Достижению метрик.

–Качеству отчета (полнота описания хода работы, анализ ошибок, визуализация).

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 экзамене;

– 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

Условия применения:

- Курс рассчитан на студентов 3-го года обучения.
- Наличие доступа к вычислительным ресурсам (GitLab).
- Инфраструктура включает индивидуальные задания с автотестами для проверки кода, интеграцию с GitLab CI/CD и облачными сервисами.

Цели, задачи и ожидаемые результаты:

– Цель:

Обеспечить студентов инструментами для работы с NLP-моделями, включая предобработку текста, обучение и развертывание моделей, а также привить навыки работы с Git и CI/CD.

– Задачи преподавателя:

1. Создание учетных записей студентов в GitLab.
2. Настройка GitLab Runner для автоматического тестирования кода.
3. Разработка шаблонного репозитория для лабораторных работ с предустановленными зависимостями (PyTorch, Transformers, spaCy).
4. Написание автотестов для проверки корректности выполнения заданий (например, токенизация, векторизация, fine-tuning моделей).
5. Визуализация результатов тестирования через HTML-отчеты.
6. Подготовка инструкций по работе с Git и облачными ресурсами.

– Ожидаемые результаты студентов:

7. Умение работать с Git, включая создание репозитория, коммиты и пул-реквесты.
8. Навыки запуска и тестирования NLP-моделей в облачных средах.
9. Понимание CI/CD-процессов в контексте NLP-разработки.

Порядок реализации:

1. Создание учетных записей в GitLab:

Каждый студент получает доступ к приватному репозиторию для выполнения заданий.

2. Настройка GitLab Runner:

Для автоматического тестирования кода используется Docker-образ с предустановленными библиотеками (PyTorch, Hugging Face Transformers).

3. Шаблонный репозиторий:

Включает:

- .gitlab-ci.yml для CI/CD.
- Скрипты для предобработки текста и обучения моделей.
- Примеры кода для работы с BERT, GPT и другими архитектурами.

4. Автотесты:

Проверяют корректность токенизации, векторизации (TF-IDF, Word2Vec), fine-tuning моделей (например, ассигасу на тестовом датасете).

5. Визуализация результатов:

Генерация HTML-отчетов с результатами тестирования, включая метрики качества моделей.

Порядок проверки корректности:

- Наличие Git-репозитория у всех студентов.
- Шаблонный репозиторий с подключенными автотестами.
- Инструкция по работе с Git и CI/CD в формате README.md.

4.4. Методические указания по организации лабораторных работ

Условия применения:

- Курс рассчитан на студентов 3-го года обучения.
- Наличие доступа к GPU/TPU (Kaggle, локальные серверы).
- Используются библиотеки: Hugging Face Transformers, spaCy, NLTK, Gensim.

Цели, задачи и ожидаемые результаты:

– Цель:

Практическое освоение методов NLP: от предобработки текста до обучения и развертывания моделей.

– Задачи преподавателя:

1. Разработка плана лабораторных работ.
2. Подготовка индивидуальных заданий (например, fine-tuning BERT для классификации текста).
3. Организация Git-инфраструктуры и автотестов.
4. Адаптация заданий под облачные среды.

– Ожидаемые результаты студентов:

1. Умение применять NLP-модели для реальных задач (классификация, генерация текста).
2. Навыки работы с Hugging Face, spaCy и другими инструментами.

Порядок реализации:

1. План лабораторных работ:

Включает:

- Предобработку текста (токенизация, лемматизация).
- Векторизацию (TF-IDF, Word2Vec).
- Обучение моделей (RNN, Transformer, BERT).
- Оценку качества (BLEU, ROUGE).

- Развертывание моделей (FastAPI, Docker).

2. Пример индивидуального задания:

Тема: Fine-tuning BERT для анализа тональности отзывов.

Технологии: Python, Hugging Face Transformers, PyTorch.

Этапы:

- Загрузка датасета (например, IMDB Reviews).
- Токенизация текста с помощью BERT-токенизатора.
- Обучение модели на GPU.
- Оценка точности на тестовой выборке.
- Развертывание модели через FastAPI.

3. Автотесты:

Проверяют:

- Корректность токенизации.
- Точность модели на валидационном датасете.
- Работоспособность API.

Порядок проверки корректности:

- Наличие выполненных заданий в Git-репозиториях.
- Прохождение автотестов.
- Отчеты с анализом результатов.

4.5. Методические указания по организации проектной деятельности студентов

Условия применения:

- Курс рассчитан на студентов 3-го года обучения.
- Время на проект: до 20 часов на студента.
- Используются кейсы от промышленных партнеров (Сбербанк, AVA LAB).

Цели, задачи и ожидаемые результаты:

- **Цель:**

Решение реальных задач NLP, таких как генерация текста, классификация, извлечение информации.

– **Задачи преподавателя:**

1. Сбор кейсов.
2. Формирование ТЗ для проектов.
3. Интеграция результатов проекта в итоговую оценку.

– **Ожидаемые результаты студентов:**

1. Реализация end-to-end NLP-решения (например, чат-бот, анализатор тональности).
2. Навыки работы в команде и презентации результатов.

Порядок реализации:

1. **Примеры кейсов:**

– **Генерация инвестиционных обзоров (Сбербанк):**

Использование GPT-3 для создания аналитических отчетов на основе структурированных данных.

– **Автоматизация обработки юридических документов (AVA LAB):**

Fine-tuning BERT для извлечения ключевых условий из договоров.

2. **ТЗ для проекта:**

Задача: Разработка NLP-пайплайна для анализа тональности отзывов.

Этапы:

- Сбор и предобработка данных.
- Обучение модели (BERT, GPT).
- Развертывание через FastAPI.
- Оценка качества (F1-score, BLEU).

3. **Критерии оценки:**

- Качество кода (прохождение автотестов).
- Точность модели.
- Документация и презентация.

Порядок проверки корректности:

- Наличие реализованного проекта в Git-репозитории.
- Отчет с анализом результатов.
- Презентация проекта на экзамене.

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

5.1 Основная литература:

1. Богданов, А. В. Современные методы обработки естественного языка : учебное пособие / А. В. Богданов. – Москва : ДМК Пресс, 2023. – 342 с. – ISBN 978-5-93700-123-4.
2. Иванов, П. К. Глубокое обучение для NLP: от RNN до Transformer / П. К. Иванов, Е. А. Смирнова. – Санкт-Петербург : БХВ-Петербург, 2022. – 256 с. – ISBN 978-5-9775-0987-6.
3. Кузнецов, М. А. Языковые модели и их применение : учебник / М. А. Кузнецов. – Москва : Лань, 2023. – 415 с. – ISBN 978-5-8114-4567-8.
4. Петров, Д. С. NLP на Python: от классики до нейросетей / Д. С. Петров. – Москва : Питер, 2022. – 320 с. – ISBN 978-5-4461-2345-6.
5. Соколова, Е. В. Библиотеки для обработки текста: spaCy, Hugging Face / Е. В. Соколова. – Москва : ДМК Пресс, 2023. – 180 с. – ISBN 978-5-93700-145-6.
6. 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
7. 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
8. Bai J. et al. From platform to knowledge graph: evolution of laboratory automation //JACS Au. – 2022. – Т. 2. – №. 2. – С. 292-309.
9. Giglio L. et al. Diverging neural dynamics for syntactic structure building in naturalistic speaking and listening //Proceedings of the National Academy of Sciences. – 2024. – Т. 121. – №. 11. – С. e2310766121.
10. Zhang Y. et al. A comprehensive large-scale biomedical knowledge graph for AI-powered data-driven biomedical research //Nature Machine Intelligence. – 2025. – С. 1-13.
11. Документация по библиотекам NLP:
 - spaCy: <https://spacy.io/usage>
 - Hugging Face Transformers: <https://huggingface.co/docs/transformers/index>
 - NLTK: <https://www.nltk.org/>

5.2 Дополнительная литература:

1. Васильев, А. Н. Трансформеры и BERT: архитектура и применение / А. Н. Васильев. – Новосибирск : Научная книга, 2023. – 210 с. – ISBN 978-5-6047890-3-2.
2. Громов, К. Л. Генеративные модели в NLP: GPT и не только / К. Л. Громов. – Москва : Интернет-Университет, 2022. – 275 с. – ISBN 978-5-9556-0123-4. Лебедев, С. А. Этика искусственного интеллекта: NLP и большие языковые модели / С. А. Лебедев. – Санкт-Петербург : Альфа-книга, 2023. – 150 с. – ISBN 978-5-98281-456-9.

3. Тихонов, Р. В. Оптимизация NLP-моделей: квантование, дистилляция / Р. В. Тихонов. – Москва : Техносфера, 2022. – 195 с. – ISBN 978-5-94836-678-1.
4. Polykovskiy, Daniil, et al. "Molecular sets (MOSES): a benchmarking platform for molecular generation models." *Frontiers in pharmacology* 11 (2020): 565644.
5. Khrabrov, Kuzma, et al. " ∇^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.

5.3. Периодические издания:

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

Конференции A*:

1. <https://openreview.net/forum?id=FMMF1a9iFC>
2. <https://openreview.net/forum?id=EIUrNM9U8c#discussion>
3. <https://openreview.net/forum?id=JoO6mtCLHD>
4. <https://aclanthology.org/2024.findings-emnlp.760/>
5. <https://aclanthology.org/2020.coling-main.588/>
6. https://link.springer.com/chapter/10.1007/978-3-030-72113-8_30
7. https://link.springer.com/chapter/10.1007/978-3-031-42448-9_10
8. <https://aclanthology.org/2024.findings-naacl.288/>

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. КиберЛенинка <http://cyberleninka.ru/>;
2. Американская патентная база данных <http://www.uspto.gov/patft/>
3. Министерство науки и высшего образования Российской Федерации <https://www.minobrnauki.gov.ru/>;
4. Федеральный портал "Российское образование" <http://www.edu.ru/>;
5. Информационная система "Единое окно доступа к образовательным ресурсам" <http://window.edu.ru/>;
6. Единая коллекция цифровых образовательных ресурсов <http://school-collection.edu.ru/> .
7. Проект Государственного института русского языка имени А.С. Пушкина "Образование на русском" <https://pushkininstitute.ru/>;
8. Справочно-информационный портал "Русский язык" <http://gramota.ru/>;
9. Служба тематических толковых словарей <http://www.glossary.ru/>;
10. Словари и энциклопедии <http://dic.academic.ru/>;
11. Образовательный портал "Учеба" <http://www.ucheba.com/>;
12. 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 курсу предусмотрено проведение лекционных занятий, на которых дается основной систематизированный материал по обработке естественного языка. В ходе лекций разбираются ключевые концепции NLP: от лингвистических основ и предобработки текста до современных архитектур нейронных сетей (RNN, Transformer, BERT). Особое внимание уделяется практическим аспектам - разбираются примеры применения NLP-технологий в реальных задачах (анализ тональности, машинный перевод, генерация текста), анализируются типичные ошибки при работе с языковыми моделями. После каждой лекции рекомендуется выполнить практические задания по разбору и обработке текстовых данных.

Лабораторные занятия курса посвящены практическому освоению методов обработки естественного языка. На занятиях детально разбираются готовые примеры реализации NLP-пайплайнов: от токенизации и нормализации текста до обучения и тонкой настройки языковых моделей. Студенты работают с реальными датасетами (отзывы,

новостные тексты, диалоги), применяя такие инструменты как NLTK, spaCy, Hugging Face Transformers. После каждого лабораторного занятия предлагаются задания для самостоятельного закрепления материала - реализация отдельных этапов обработки текста или модификация рассмотренных алгоритмов.

При самостоятельной работе студентам необходимо изучать рекомендованную литературу (учебники, научные статьи, документацию библиотек) для глубокого понимания теоретических основ и современных подходов в NLP. Выполняя проектные задания, студент должен уметь: формулировать задачу обработки текста (классификация, генерация, извлечение информации); подбирать и предобрабатывать данные; выбирать и реализовывать подходящие методы (от классических алгоритмов до нейросетевых архитектур); оценивать качество работы модели с помощью соответствующих метрик. Особое внимание уделяется навыкам отладки и оптимизации NLP-моделей.

Важнейшим компонентом курса является самостоятельная проектная работа, в ходе которой студент разрабатывает законченное NLP-приложение (например, чат-бот, анализатор текстов или система машинного перевода). Такой проект позволяет закрепить навыки проектирования и реализации комплексных решений в области обработки естественного языка.

Для студентов с ограниченными возможностями здоровья предусмотрены дополнительные индивидуальные консультации, на которых преподаватель детально разъясняет сложные аспекты дисциплины, помогает адаптировать задания и обеспечивает специальные условия для освоения практических навыков работы с текстовыми данными. Индивидуальный подход позволяет таким студентам полноценно участвовать в учебном процессе и достигать требуемых результатов обучения.

Кейсы ПАО «Сбербанк»

1. Генеративный ИИ для автоматического составления инвестиционных обзоров

Описание:

Аналитики Сбера ежедневно составляют десятки аналитических и инвестиционных обзоров по рынкам, компаниям, макроэкономике. Задача — исследовать применение LLM для генерации кратких сводок и аналитических отчетов на основе входных данных: биржевые котировки, макроэкономические показатели, рыночные события.

Цель:

Разработать инструмент, способный по структурированным данным и краткому описанию формировать инвестиционный обзор в деловом стиле.

Ожидаемый результат:

Модель, генерирующая аналитические тексты длиной 500–1000 слов с разделами «обзор событий», «рекомендации», «прогнозы», оформленные в формате банка.

2. Генерация сценариев фишинговых писем для обучения сотрудников

Описание:

Банк проводит киберучения, включая рассылку тестовых фишинговых писем сотрудникам для повышения их устойчивости к социальным атакам. Проект предполагает использование генеративной модели для создания реалистичных фишинговых писем различных типов (поддельные счета, HR-запросы, ИТ-поддержка).

Цель:

Создать генератор, способный на основе заданных параметров (тема, стиль, уровень угрозы) создавать тексты фишинга для тренировок.

Ожидаемый результат:

Набор разнообразных примеров фишинга и оценка их эффективности по реакции сотрудников, а также классификация моделей угроз.

Кейсы от «АВАЛАБ»

3. Генерация рекламного контента для жилых комплексов

Описание:

«АВА ГРУПП» регулярно запускает маркетинговые кампании для жилых комплексов. Необходимо исследовать использование диффузионных моделей для генерации изображений (визуализации интерьеров, окрестностей, видов из окон) и LLM — для описаний квартир, преимуществ района, инфраструктуры.

Цель:

Создать инструменты для быстрой генерации продающих материалов без привлечения дизайнеров и копирайтеров на первых этапах.

Ожидаемый результат:

Набор сгенерированных карточек объектов с текстом, изображением и логикой «живого» рекламного сообщения.

4. Генерация документации и шаблонов договоров

Описание:

Юридический департамент регулярно работает с договорами долевого участия, актами приёма-передачи и другими документами. Использование LLM может значительно сократить время на подготовку черновиков — достаточно ввести параметры сделки.

Цель:

Создать систему, которая генерирует адаптированные тексты документов по вводным данным (тип объекта, этаж, площадь, ФИО, сроки и пр.).

Ожидаемый результат:

Генератор документов в формате Word или PDF с автоматической подстановкой параметров и соблюдением юридического стиля.

5. Обратная генерация — ИИ-помощник для покупателей квартир

Описание:

Будущие покупатели часто задают типовые вопросы о квартирах, планировках, ипотеке, акциях, сроках. Вместо call-центра предлагается реализовать LLM-бота, который обрабатывает текстовые и голосовые запросы, показывает планировки, ссылается на PDF-документы и может «объяснять» информацию простым языком.

Цель:

Упростить коммуникацию с клиентами на этапе выбора квартиры и повысить качество первичного контакта.

Ожидаемый результат:

Демо-бот, способный отвечать на вопросы о жилом комплексе, ориентируясь в его характеристиках и маркетинговых документах.

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

7.1 Перечень информационно-коммуникационных технологий

1. Облачные платформы и сервисы

Google Colab – облачная среда для выполнения кода на Python с GPU/TPU

Kaggle – платформа для работы с датасетами и соревнований по ML

Hugging Face Spaces – развертывание NLP-моделей в виде демо

AWS/GCP/Azure/YandexCloud – облачные вычисления для обучения и инференса моделей

2. Системы управления версиями и коллаборации

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

Notion/Trello – организация проектной деятельности

3. Инструменты для работы с данными

Label Studio – разметка датасетов

DVC (Data Version Control) – управление версиями данных

4. Коммуникационные технологии

Telegram – координация работы в команде

МТС Линк– проведение онлайн-консультаций и защиты проектов

7.2 Перечень лицензионного и свободно распространяемого программного обеспечения

1. Лицензионное ПО

VSCoде – IDE для Python (свободнораспространяемое)

LibreOffice– оформление отчетов (свободнораспространяемое)

2. Свободное ПО (Open Source)

Библиотеки для NLP:

NLTK – обработка текста

sраСу – промышленные NLP-пайплайны

Hugging Face Transformers – предобученные модели (BERT, GPT)

Gensim – тематическое моделирование и word2vec

Фреймворки для ML:

PyTorch/TensorFCow – 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)

№	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.