Обработка текста на естественном языке (NLP)
Лекция 1: задачи, пайплайн, предобработка и инструменты

Преподаватель: Четвергов Андрей Сергеевич

РАНХиГС — ИЭМИТ
Кафедра эконометрики и математической экономики

Весенний семестр 2026

Структура лекции

1. О курсе: формат и ожидания
2. Зачем NLP и что мы решаем
3. Пайплайн: от текста к модели
4. Предобработка и токенизация
5. Представления, модели, метрики
6. Инструменты, контроль, следующий шаг

Навигация по лекции

1. О курсе: формат и ожидания
2. Зачем NLP и что мы решаем
3. Пайплайн: от текста к модели
4. Предобработка и токенизация
5. Представления, модели, метрики
6. Инструменты, контроль, следующий шаг

О курсе: формат и ожидания

Формат

- 5 лекций + 8 семинаров (практика в ноутбуках)
- 2 домашних задания + проект
- Экзамен: устное собеседование по вашим работам

Что считается “хорошей сдачей”

- воспроизводимый код (ноутбук/репо)
- таблица метрик + сравнение
- примеры ошибок + объяснение причин
- короткие, ясные выводы

Как будет устроена работа

- на семинарах — пишем пайплайн и учимся считать метрики
- в домашних — сравниваем подходы и делаем выводы по ошибкам

Правило курса

Сначала сильный baseline (TF-IDF + линейная модель), затем улучшения по результатам ошибок.

Навигация по лекции

1. О курсе: формат и ожидания
2. Зачем NLP и что мы решаем
3. Пайплайн: от текста к модели
4. Предобработка и токенизация
5. Представления, модели, метрики
6. Инструменты, контроль, следующий шаг

Зачем нужен NLP в прикладной аналитике

Текст — один из самых частых и самых “шумных” источников данных.

Экономика и бизнес: типовые кейсы

- Отзывы/обращения: тональность, причины недовольства, драйверы качества
- Поддержка: категоризация и маршрутизация тикетов
- Новости: мониторинг событий и рисков, упоминания компаний/персон
- Документы: извлечение дат, сумм, контрагентов, ключевых фактов

Что даёт NLP

- автоматизация рутинных разборов текста
- анализ на масштабе (тысячи/миллионы сообщений)
- прозрачная оценка качества (метрики + ошибки)

Цель курса

Научиться строить работающие NLP-пайплайны и корректно оценивать качество.

Типовые задачи NLP (карта)

Классификация

- категория обращения (billing/delivery/tech)
- тема новости / рубрика документа
- спам / не спам

Sequence labeling

- POS-tagging (части речи)
- NER: персоны, компании, даты, суммы

Тональность (Sentiment)

- позитив / негатив / нейтрально
- “почему негатив?” — причины и триггеры

Сходство / поиск

- похожие документы, дедупликация
- кластеризация текстов

Траектория курса: от классики (TF-IDF) к современным моделям (эмбеддинги/трансформеры).

Эры NLP: что было “в моде” и почему

Очень грубая, но полезная “карта эпох”

- 1950s–1980s: правила и лингвистика — грамматики, словари, шаблоны; много ручной инженерии.
- 1990s–2012: статистическая эпоха — n-граммные языковые модели, HMM/CRF, SVM; TF-IDF как рабочий baseline.
- 2013–2017: нейросети до трансформеров — word2vec/GloVe, RNN/LSTM/GRU; рост качества в последовательных задачах.
- 2018–2020: трансформерная революция — BERT/GPT, self-attention, transfer learning; fine-tuning под задачу.
- 2020–н.в.: foundation/LLM эпоха — GPT-3/4-подобные, instruction tuning, RLHF, RAG; zero/few-shot и “модели как сервис”.

Позиция курса

Мы начинаем с классики (TF-IDF + линейные модели) как честного baseline, затем переходим к эмбеддингам/трансформерам.

Почему язык сложнее табличных данных

Неоднозначность

- контекст, омонимия, сарказм
- разные формулировки одной мысли

Шум и вариативность

- опечатки, сленг, сокращения
- разные жанры: отзывы, новости, документы

Русский язык

- богатая морфология: формы слов, согласования
- свободный порядок слов и “дальние” связи

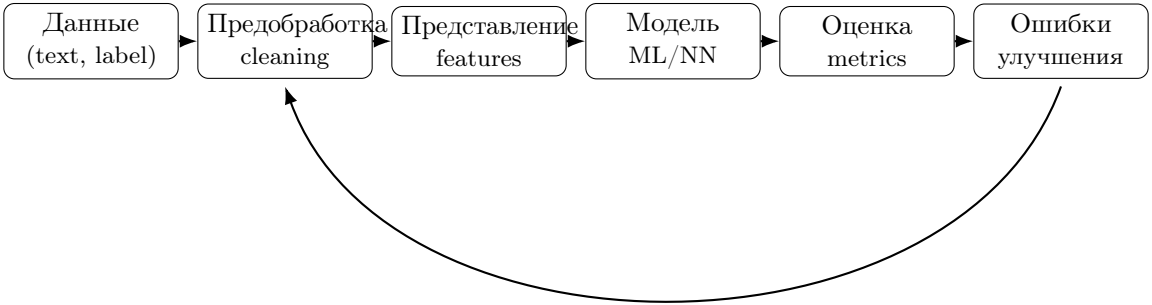
Практический вывод

Модель должна быть устойчивой к шуму, а метрики — отражать реальную цель задачи.

Навигация по лекции

1. О курсе: формат и ожидания
2. Зачем NLP и что мы решаем
3. Пайплайн: от текста к модели
4. Предобработка и токенизация
5. Представления, модели, метрики
6. Инструменты, контроль, следующий шаг

Базовый пайплайн NLP: от текста к результату



- Этот “скелет” повторяется в ДЗ1 и проекте.
- Улучшения делаем через анализ ошибок, а не “усложнение ради усложнения”.

Как выглядит “хорошее решение”

Мини-чеклист (то, что мы будем требовать в ДЗ)

- чёткая постановка задачи + выбранная метрика (что оптимизируем?)
- честный baseline (TF-IDF + линейная модель)
- корректное разделение данных (train/valid/test)
- таблица метрик + сравнение подходов
- примеры ошибок + гипотезы улучшений

Главная идея

Качество = не “красивая модель”, а воспроизводимый пайплайн и объяснимые улучшения.

Навигация по лекции

1. О курсе: формат и ожидания
2. Зачем NLP и что мы решаем
3. Пайплайн: от текста к модели
4. Предобработка и токенизация
5. Представления, модели, метрики
6. Инструменты, контроль, следующий шаг

Предобработка: что делаем и зачем

Цель

Снизить шум и привести тексты к более стабильному виду без потери смысла.

Типовые шаги

- нормализация: lower(), пробелы, типографика
- очистка: HTML/спецсимволы (по задаче)
- токенизация: слова / подслова
- стоп-слова (осторожно)
- лемматизация vs стемминг (особенно для русского)

Частая ошибка

“Перечистить” текст так, что исчезает важная информация: отрицание, числа, имена, единицы измерения.

Правило

Для трансформеров часто достаточно минимальной очистки.

Токенизация: слова, символы, подслова

Зачем токенизация

- модель работает с последовательностью токенов
- от токенов зависит словарь и размерность признаков

Варианты

- word-level: слова (классика, удобно для TF-IDF)
- char-level: символы (устойчиво к опечаткам, но длинно)
- subword-level: BPE/WordPiece (стандарт для BERT)

Интуиция subword

Подслова помогают с редкими словами и морфологией:
“эконометрический” $\approx$ “эконометр” + “ический”

В курсе

TF-IDF (слова/n-граммы) + токенизация BERT (подслова).

Стоп-слова, стемминг, лемматизация: когда полезно

Стоп-слова

- уменьшают размерность и шум
- но могут “сломать” смысл (частицы, отрицание)

Стемминг vs лемматизация

- стемминг: быстро, грубо (обрезает основу)
- лемматизация: аккуратнее, но тяжелее

Пример риска

“не понравилось” $\rightarrow$ “понравилось” (если удалить “не”)

Практический критерий

Для русского и задач “качество важнее скорости” — чаще берут лемматизацию.

Навигация по лекции

1. О курсе: формат и ожидания
2. Зачем NLP и что мы решаем
3. Пайплайн: от текста к модели
4. Предобработка и токенизация
5. Представления, модели, метрики
6. Инструменты, контроль, следующий шаг

TF, IDF и TF-IDF: что это и откуда берётся

Идея

Слова, которые часто встречаются в документе, но редко в коллекции, более информативны.

TF (term frequency)

$$\mathrm{tf}(t, d) = \frac{f_{t,d}}{\sum_{t'} f_{t',d}}$$

где $f_{t,d}$ — сколько раз термин t встретился в документе d.

IDF (inverse document frequency)

$$\mathrm{idf}(t) = \log \frac{N + 1}{\mathrm{df}(t) + 1} + 1$$

N — число документов, $\mathrm{df}(t)$ — сколько документов содержит t.

TF-IDF (итоговый вес)

$$\mathrm{tfidf}(t, d) = \mathrm{tf}(t, d) \cdot \mathrm{idf}(t)$$

Представления текста: что выбрать

BoW / TF-IDF / n-граммы

- быстрые и сильные baseline'ы
- хороши на небольших датасетах
- интерпретируемость: важные слова/фразы

Эмбеддинги / трансформеры

- лучше ловят семантику и контекст
- часто выигрывают на "тонких" задачах
- режимы:
  - feature-based: эмбеддинги → классификатор
  - fine-tuning: дообучение под задачу

Позиция курса

Сначала baseline на TF-IDF, затем сравниваем с эмбеддингами/BERT.

Модели, которые будем использовать

Классические модели (для TF-IDF/BoW)

- Naive Bayes — быстрый baseline
- Logistic Regression — часто очень сильная на тексте
- Linear SVM — мощная линейная модель

Контекстные модели

- sequence labeling (задачи разметки: NER и др.)
- Transformer / BERT-подобные модели (через HuggingFace)

Важно

Усложнять модель имеет смысл только после корректной метрики и baseline.

Метрики качества: что и когда смотреть

Классификация

- Accuracy — осторожно при дисбалансе
- Precision / Recall — “ложные срабатывания” vs “пропуски”
- F1 — часто базовая метрика для текста

Кривые

- ROC-AUC — общая разделимость
- PR-AUC — предпочтительно при дисбалансе

Анализ ошибок

- confusion matrix (где именно ошибаемся)
- примеры конкретных ошибочных текстов
- идеи улучшений: данные / признаки / порог / модель

Принцип

Метрика без ошибок — “цифра без понимания”.

Навигация по лекции

1. О курсе: формат и ожидания
2. Зачем NLP и что мы решаем
3. Пайплайн: от текста к модели
4. Предобработка и токенизация
5. Представления, модели, метрики
6. Инструменты, контроль, следующий шаг

Python-экосистема курса (минимальный стек)

- данные: pandas, numpy
- предобработка: re, nltk, spaCy, natasha/pymorphy2
- признаки + ML: scikit-learn (CountVectorizer, TF-IDF, модели, метрики)
- современные модели: transformers (HuggingFace), иногда datasets

Результат каждой практики/ДЗ

Один ноутбук (или репозиторий) + таблица метрик + выводы + примеры ошибок.

Мини-кейс: классификация отзывов (шаблон решения)

Задача

Дано: тексты отзывов + метка (позитив/негатив). Нужно сравнить подходы.

Пайплайн

1. train/test split
2. минимальная предобработка
3. TF-IDF (и/или n-граммы)
4. модель: LogReg / NB / Linear SVM
5. метрики: F1 + confusion matrix

Что обязательно в выводах

- почему выбранная модель лучше
- на каких текстах она ошибается
- что бы вы улучшали дальше

Следующее занятие: семинар 1

Подготовка к семинару

- окружение: conda/venv
- установить: pandas, numpy, scikit-learn
- установить: nltk, spacy, natasha
- (опционально) transformers

На семинаре 1 сделаем

предобработку + TF-IDF baseline + первые метрики + разбор ошибок на примерах.

Вопросы?