# ЛЕКЦИЯ 9
# МЕТОДЫ ОБРАБОТКИ ЕСТЕСТВЕННОГО ЯЗЫКА

**Цель лекции**

Изучить основные понятия и методы обработки естественного языка (Natural Language Processing, NLP), этапы подготовки текстовых данных, токенизацию и нормализацию текста, способы векторного представления документов, методы классификации и анализа тональности, извлечение именованных сущностей, семантический поиск и машинный перевод; рассмотреть практические примеры применения NLP в интеллектуальных системах и информационной безопасности.

**Понятие обработки естественного языка**

***Обработка естественного языка (Natural Language Processing, NLP)*** — направление искусственного интеллекта, связанное с компьютерной обработкой, анализом, пониманием и генерацией человеческого языка. Текст является неструктурированным типом данных, поэтому перед применением алгоритмов машинного обучения его необходимо преобразовать в форму, пригодную для вычислительной обработки.

К задачам NLP относятся классификация текстов, анализ тональности, извлечение информации, поиск именованных сущностей, тематический анализ, автоматическое реферирование, семантический поиск, ответы на вопросы и машинный перевод.

| **Задача NLP** | **Входные данные** | **Результат** |
|---|---|---|
| Классификация текста | Сообщение, документ, обращение | Категория текста |
| Анализ тональности | Отзыв или комментарий | Положительная, отрицательная или нейтральная оценка |
| NER | Текст документа | Имена, организации, даты, географические объекты |
| Суммаризация | Длинный документ | Краткое содержание |
| Семантический поиск | Запрос и коллекция документов | Наиболее смыслово близкие документы |
| Машинный перевод | Текст на одном языке | Текст на другом языке |

**Особенности естественного языка**

Естественный язык сложен для автоматической обработки, поскольку одно слово может иметь несколько значений, смысл зависит от контекста, а одинаковая мысль может выражаться разными словами и грамматическими конструкциями. Дополнительные трудности создают опечатки, сокращения, разговорные формы, специальные термины и смешение языков.

***Например***, слово «ключ» может обозначать инструмент, источник воды или криптографический ключ. Определить нужное значение можно только с учётом окружающего контекста. Поэтому современные NLP-системы стремятся учитывать не только отдельные слова, но и связи между ними.

| **Особенность** | **Пример** | **Проблема для модели** |
|---|---|---|
| Многозначность | ключ шифрования / гаечный ключ | Требуется контекст |

| | | |
|---|---|---|
| Синонимия | атака / нападение | Разные слова могут иметь близкий смысл |
| Морфология | документ, документа, документами | Разные формы одной лексемы |
| Опечатки | авторизацыя | Усложняют словарное сопоставление |
| Многоязычие | құжат / документ / document | Необходима языковая адаптация |

**Общая схема обработки текста**

Типичный NLP-конвейер включает получение текстовых данных, очистку, токенизацию, нормализацию, преобразование текста в числовое представление, обучение или применение модели и оценку результата. Конкретный набор этапов зависит от задачи и используемой модели.

| Этап | Назначение | Пример |
|---|---|---|
| Сбор текста | Формирование корпуса | Отзывы, журналы событий, документы |
| Очистка | Удаление технического шума | HTML-разметка, лишние пробелы |
| Токенизация | Разделение текста на единицы | слова, подслова, предложения |
| Нормализация | Приведение вариантов к сопоставимому виду | регистр, лемматизация |
| Векторизация | Преобразование текста в числа | TF-IDF, embeddings |
| Моделирование | Решение целевой задачи | классификация, NER, перевод |
| Оценка | Проверка качества | precision, recall, F1 и другие метрики |

**Очистка текстовых данных**

Очистка текста устраняет элементы, которые не несут полезной информации для конкретной задачи или создают технический шум. Возможные операции включают удаление лишних пробелов, служебных символов, HTML-тегов, повторяющихся знаков, приведение регистра и обработку ссылок.

Однако очистка не должна выполняться механически. Например, знаки пунктуации могут быть важны для анализа тональности, а регистр — для распознавания именованных сущностей. Поэтому набор операций определяется целью исследования.

**Токенизация**

***Токенизация*** — разбиение текста на отдельные единицы, называемые токенами. Токенами могут быть слова, части слов, символы или предложения. Это один из базовых этапов обработки текста.

***Пример***: предложение «Модель обнаружила подозрительное соединение» при простой токенизации по словам может быть представлено как [«Модель», «обнаружила», «подозрительное», «соединение»].

| Вид токенизации | Единица | Применение |
|---|---|---|
| По предложениям | Предложение | Анализ структуры документа |
| По словам | Слово | Классические NLP-задачи |
| Подсловная | Часть слова | Современные нейросетевые языковые модели |
| Посимвольная | Символ | Специализированные задачи, устойчивость к неизвестным словам |

**Нормализация текста**

После токенизации текст может быть нормализован. Нормализация уменьшает количество различных форм, которые модель должна учитывать. К распространённым операциям относятся приведение к нижнему регистру, удаление или специальная обработка пунктуации, лемматизация и стемминг.

Лемматизация приводит слово к словарной форме с учётом морфологии. Например, «документы», «документа» и «документами» могут быть приведены к лемме «документ». Стемминг обычно механически сокращает слово до основы и может давать форму, не являющуюся полноценным словом.

| Метод | Идея | Особенность |
|---|---|---|
| Lowercase | Приведение к одному регистру | Уменьшает число вариантов, но может потерять информацию о регистре |
| Стемминг | Отсечение частей слова | Быстрее, но грубее |
| Лемматизация | Получение словарной формы | Требует языковых правил или модели |
| Удаление стоп-слов | Исключение частых служебных слов | Полезно не для всех задач |

**Стоп-слова**

***Стоп-слова*** — часто встречающиеся слова, которые в некоторых задачах имеют небольшую различительную ценность. К ним могут относиться союзы, предлоги и некоторые служебные слова. В классических моделях их иногда удаляют для уменьшения размерности.

В современных контекстных моделях стоп-слова часто сохраняют, поскольку они участвуют в формировании смысла и грамматической структуры предложения. Следовательно, удаление стоп-слов не является обязательным правилом.

**Мешок слов — Bag of Words**

Bag of Words представляет документ через частоты слов из заранее сформированного словаря. Порядок слов обычно не учитывается. Если словарь содержит m терминов, каждый документ превращается в вектор длины m.

***Пример***. Для словаря [«атака», «сервер», «доступ»] документ «атака на сервер» можно представить вектором [1, 1, 0]. Такой подход прост и интерпретируем, но плохо учитывает контекст и семантику.

**TF-IDF**

***TF-IDF*** — способ числового представления текста, который учитывает частоту термина в документе и его распространённость во всей коллекции. Термин получает больший вес, если он часто встречается в конкретном документе, но сравнительно редко встречается в других документах.

Упрощённо TF-IDF можно представить как произведение двух величин:

$$\text{TF-IDF}(t, d) = \text{TF}(t, d) \times \text{IDF}(t)$$

TF показывает частоту термина t в документе d, а IDF уменьшает вес слов, встречающихся во многих документах. В результате более специфичные для документа термины получают больший вес.

| **Подход** | **Учитывает частоту** | **Учитывает редкость слова в корпусе** | **Учитывает контекст** |
|---|---|---|---|
| Bag of Words | Да | Нет | Нет |
| TF-IDF | Да | Да | Нет |
| Контекстные embeddings | Косвенно | Не как отдельную формулу | Да |

**Пример TF-IDF в Python**

```
from sklearn.feature_extraction.text import TfidfVectorizer

texts = [
    'подозрительная попытка входа в систему',
    'успешный вход пользователя в систему',
    'обнаружена подозрительная сетевая активность'
]

vectorizer = TfidfVectorizer()
X = vectorizer.fit_transform(texts)

print(vectorizer.get_feature_names_out())
print(X.toarray())
```

В результате каждый текст преобразуется в числовой вектор. Полученную матрицу можно передать алгоритму классификации, кластеризации или поиска сходства.

**Векторные представления слов**

Ограничение Bag of Words и TF-IDF состоит в том, что смысловые отношения между словами напрямую не моделируются. Векторные представления слов (word embeddings) отображают слова в плотное числовое пространство, где семантически близкие слова могут иметь близкие векторы.

В загруженных материалах среди методов NLP указаны Word2Vec и GloVe. Такие подходы позволяют перейти от разреженных частотных векторов к плотным представлениям, отражающим статистические отношения между словами.

| Представление | Тип вектора | Сильная сторона | Ограничение |
|---|---|---|---|
| TF-IDF | Разреженный | Интерпретируемость и простота | Не учитывает контекст слова |
| Word2Vec | Плотный | Семантическая близость слов | Обычно одно представление слова независимо от контекста |
| GloVe | Плотный | Использует глобальную статистику совместной встречаемости | Контекст конкретного употребления ограничен |
| BERT-представления | Контекстные | Учитывают окружение слова | Требуют больше вычислительных ресурсов |

**Контекстные языковые представления**

Современные нейросетевые методы формируют представление слова с учётом окружающего текста. Поэтому одно и то же слово в разных предложениях может иметь разные векторы. В загруженных материалах среди NLP-подходов указаны BERT и GPT-модели, способные лучше учитывать контекст и семантические связи, но предъявляющие более высокие требования к вычислительным ресурсам.

***Например***, значение слова «ключ» в выражениях «ключ шифрования» и «ключ от двери» различается. Контекстная модель стремится сформировать разные представления этих употреблений.

**Классификация текстов**

***Классификация текста*** — задача присвоения документу одной или нескольких категорий. Классический подход включает очистку текста, TF-IDF-векторизацию и обучение классификатора, например логистической регрессии, SVM или Naive Bayes.

***Примеры***: классификация обращений пользователей по темам; определение фишинговых сообщений; распределение документов по категориям; классификация событий информационной безопасности по типу инцидента.

| Этап | Пример для фишинговых сообщений |
|---|---|
| Данные | Набор сообщений с метками phishing / legitimate |
| Предобработка | Очистка и токенизация |
| Векторизация | TF-IDF |
| Модель | Logistic Regression или SVM |
| Оценка | Precision, recall, F1, confusion matrix |

**Пример классификации текста**

```
from sklearn.model_selection import train_test_split
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import classification_report

X_train, X_test, y_train, y_test = train_test_split(
    texts, labels, test_size=0.2, random_state=42, stratify=labels
)
```

```
vectorizer = TfidfVectorizer(ngram_range=(1, 2))
X_train_vec = vectorizer.fit_transform(X_train)
X_test_vec = vectorizer.transform(X_test)

model = LogisticRegression(max_iter=1000)
model.fit(X_train_vec, y_train)
y_pred = model.predict(X_test_vec)

print(classification_report(y_test, y_pred))
```

**N-граммы**

***N-грамма*** — последовательность из n соседних токенов. Unigram содержит один токен, bigram — два, trigram — три. N-граммы позволяют частично учитывать локальный порядок слов в классических векторных моделях.

***Например***, слова «не» и «безопасно» по отдельности могут не передавать полный смысл, а bigram «не безопасно» содержит более информативную комбинацию. Однако увеличение числа n резко увеличивает размер словаря.

**Анализ тональности**

***Анализ тональности*** (sentiment analysis) определяет эмоциональную или оценочную окраску текста. Базовая постановка включает положительный, отрицательный и нейтральный классы, хотя в прикладных системах возможны более сложные шкалы.

***Например***, отзывы пользователей о цифровом сервисе можно автоматически разделить на положительные и отрицательные, а затем анализировать наиболее частые причины недовольства. Для юридических и технических документов термин «тональность» может быть менее естественным, поэтому постановка задачи должна соответствовать типу текста.

**Извлечение именованных сущностей NER**

Named Entity Recognition (NER) — выделение в тексте сущностей определённых типов. К типичным категориям относятся люди, организации, географические объекты, даты и другие предметно-ориентированные сущности.

В юридическом документе NER может использоваться для поиска участников, организаций, дат и других реквизитов. В журналах информационной безопасности предметно-ориентированная система может извлекать IP-адреса, домены, имена файлов и идентификаторы, если такие категории предусмотрены моделью или правилами.

| **Текстовый фрагмент** | **Возможная сущность** |
|---|---|
| Satbayev University | Организация |
| 11 сентября 2026 года | Дата |
| Алматы | Географический объект |
| example.kz | Домен в специализированной системе безопасности |

**Тематическое моделирование**

Тематическое моделирование предназначено для выявления скрытых тематических структур в коллекции документов. В загруженных материалах среди методов анализа текста указано LDA. Модель рассматривает документы как сочетания тем, а темы — как распределения слов.

Такой подход может применяться для первичного исследования больших коллекций обращений или документов, когда заранее неизвестно, какие тематические группы присутствуют в корпусе.

**Семантический поиск**

Семантический поиск стремится находить документы не только по совпадению слов, но и по смысловой близости. Для этого запрос и документы преобразуются в векторные представления, после чего вычисляется их сходство.

***Например***, запрос «несанкционированный доступ к аккаунту» может быть смыслово близок документу, содержащему выражение «взлом учётной записи», даже если точные слова не совпадают. Это важное отличие от простого лексического поиска.

**Косинусное сходство**

Для сравнения текстовых векторов часто используется косинусное сходство. Оно оценивает угол между двумя векторами и меньше зависит от их абсолютной длины:

$$\cos(x, y) = (x \cdot y) / (||x|| \, ||y||)$$

Чем ближе значение к 1 для неотрицательных текстовых представлений, тем более сходным считается направление векторов. Косинусное сходство широко применяется к TF-IDF и embedding-представлениям.

**Суммаризация текста**

Суммаризация предназначена для получения краткого представления длинного текста. Экстрактивные методы выбирают наиболее значимые фрагменты исходного документа. Абстрактивные методы формируют новое краткое изложение содержания.

| **Тип** | **Принцип** | **Особенность** |
|---|---|---|
| Экстрактивная | Выбор предложений или фрагментов из исходного текста | Сохраняет исходные формулировки |
| Абстрактивная | Генерация нового краткого текста | Может быть более естественной, но требует проверки фактической точности |

**Машинный перевод**

***Машинный перевод*** — автоматическое преобразование текста с одного естественного языка на другой. В учебном пособии машинный перевод включён в основные темы главы NLP. Современные системы перевода обычно используют нейросетевые архитектуры и обучаются на больших параллельных или многоязычных корпусах.

Для казахского, русского и английского языков особое значение имеют качество многоязычных данных, морфологические особенности языков, терминология предметной области и корректная передача именованных сущностей.

**NLP для казахского и русского языков**

Многоязычная обработка требует учитывать различия языков. Казахский язык обладает развитой словоизменительной системой, поэтому одна основа может встречаться во множестве словоформ. Для русского языка также важны морфологические формы, падежи и согласование. В смешанных сообщениях дополнительно встречается переключение между языками.

При разработке системы необходимо проверять качество токенизатора, нормализации и модели отдельно на каждом целевом языке, а не переносить результаты одного языка на другой без экспериментальной проверки.

**NLP в информационной безопасности**

| Задача | Текстовые данные | Методы NLP |
|---|---|---|
| Обнаружение фишинга | Письма и сообщения | TF-IDF/embeddings + классификация |
| Анализ журналов и SOC | Описания событий и инцидентов | Классификация, NER, семантический поиск |
| Анализ угроз | Отчёты и бюллетени | NER, извлечение информации, суммаризация |
| Поиск похожих инцидентов | История обращений и отчётов | Embeddings и семантическое сходство |
| Маршрутизация обращений | Текст заявок | Классификация по категориям |

NLP не заменяет технические средства обнаружения атак. Он дополняет их, позволяя анализировать текстовые источники, сообщения, отчёты и описания событий.

**Пример: анализ фишингового сообщения**

Рассмотрим задачу бинарной классификации электронных сообщений. Обучающая выборка содержит тексты и метки legitimate или phishing. После очистки и TF-IDF-векторизации обучается классификатор. Для оценки необходимо учитывать не только accuracy, но и precision и recall класса phishing, поскольку пропуск опасного сообщения и ложная блокировка легитимного письма имеют разную цену.

Дополнительно могут использоваться признаки URL, домена, структуры письма и метаданных. Это пример мультимодального по смыслу решения, где текстовые признаки NLP объединяются с техническими характеристиками сообщения.

**Пример: анализ юридических документов**

Для большого корпуса юридических документов NLP может применяться последовательно: токенизация и очистка текста; TF-IDF или контекстные представления; извлечение сущностей; классификация документов; тематическое моделирование; суммаризация; семантический поиск. В загруженных проектных материалах для анализа судебных данных перечислены TF-IDF, Word2Vec, GloVe, BERT и GPT-модели, а среди задач — NER, анализ тональности, LDA, суммаризация и семантический поиск.

**Оценка качества NLP-моделей**

Метрика зависит от задачи. Для классификации применяются accuracy, precision, recall и F1-score. Для NER также часто рассчитывают precision, recall и F1 по сущностям. Для машинного перевода и суммаризации используются специальные автоматические метрики, однако их необходимо дополнять содержательной оценкой качества текста.

| Задача | Примеры метрик | Что важно проверить дополнительно |
|---|---|---|
| Классификация | Precision, recall, F1 | Ошибки по каждому классу |

| | | |
|---|---|---|
| NER | Precision, recall, F1 | Правильность границ и типов сущностей |
| Семантический поиск | Recall@k, Precision@k и ранговые метрики | Релевантность найденных документов |
| Перевод | Автоматические метрики качества перевода | Смысл, терминология, грамматика |
| Суммаризация | Автоматические метрики сходства | Фактическая корректность и полнота |

**Разделение данных и утечка информации**

При обучении NLP-модели необходимо разделять данные на обучающую, валидационную и тестовую части. Особое внимание следует уделять утечке данных. Например, почти одинаковые версии одного документа не должны одновременно попадать в train и test, если это делает тест искусственно простым.

Векторизатор TF-IDF также должен обучаться только на обучающей части, а к test применяться уже настроенное преобразование. Иначе информация о тестовой коллекции частично попадает в процесс подготовки признаков.

**Классические методы и нейросетевой NLP**

| **Критерий** | **Классический NLP** | **Нейросетевой/контекстный NLP** |
|---|---|---|
| Представление | Bag of Words, TF-IDF | Embeddings, контекстные представления |
| Модели | LogReg, SVM, Naive Bayes | Нейронные сети, Transformer-модели |
| Контекст | Ограниченный | Учитывается значительно лучше |
| Вычислительные требования | Обычно ниже | Обычно выше |
| Интерпретируемость | Часто выше | Часто сложнее |
| Малые данные | Может быть очень конкурентоспособным | Часто требуется перенос обучения или готовая модель |

**Типичные ошибки при работе с текстом**

– Удалять пунктуацию, регистр или стоп-слова без проверки, нужны ли они для конкретной задачи.

– Обучать TF-IDF на всём наборе данных до разделения train/test.

– Считать TF-IDF семантическим представлением, полностью учитывающим контекст.

– Использовать только accuracy при сильном дисбалансе классов.

– Не учитывать язык, морфологию и смешение языков в многоязычном корпусе.

– Считать автоматически сгенерированную суммаризацию или перевод безусловно корректными.

– Сравнивать модели на разных тестовых выборках и делать вывод о превосходстве алгоритма.

– Использовать сложную нейросетевую модель без сравнения с простой базовой моделью.

**Практический пример полного NLP-конвейера**

Пусть необходимо автоматически классифицировать обращения пользователей информационной системы по категориям «доступ», «сетевая проблема», «подозрительная активность» и «другое». Сначала формируется размеченный корпус. Затем тексты очищаются с сохранением значимых технических обозначений, выполняется токенизация и строится TF-IDF-представление. На одинаковом разбиении данных сравниваются Logistic Regression и SVM.

После оценки confusion matrix и F1-score по каждому классу анализируются типичные ошибки. Если система плохо распознаёт смыслово близкие формулировки с разной лексикой, можно перейти к embedding-представлениям или контекстной модели. Таким образом, сложность решения увеличивается только при наличии экспериментальной необходимости.

**Основные выводы**

– NLP предназначен для компьютерной обработки, анализа, понимания и генерации естественного языка.

– Текст требует специальной подготовки и преобразования в числовое представление.

– Токенизация разделяет текст на единицы, а нормализация уменьшает вариативность форм.

– Bag of Words и TF-IDF являются классическими способами векторизации текста.

– Word2Vec и GloVe формируют плотные векторные представления слов, а контекстные модели учитывают окружение слова.

– К основным задачам NLP относятся классификация, анализ тональности, NER, тематическое моделирование, суммаризация, семантический поиск и машинный перевод.

– В информационной безопасности NLP применяется к фишинговым сообщениям, журналам, отчётам об угрозах и описаниям инцидентов.

– Качество NLP-системы необходимо оценивать на независимых данных и подходящими для задачи метриками.

**Контрольные вопросы**

1. Что понимается под обработкой естественного языка?
2. Какие основные задачи относятся к NLP?
3. Почему естественный язык сложен для автоматической обработки?
4. Что такое текстовый корпус?
5. Из каких этапов может состоять NLP-конвейер?
6. Что такое токенизация?
7. Какие виды токенизации существуют?
8. Что понимается под нормализацией текста?
9. Чем лемматизация отличается от стемминга?
10. Что такое стоп-слова и всегда ли их следует удалять?
11. Как работает Bag of Words? В чём основная идея TF-IDF?
13. Что показывает TF в TF-IDF? Какова роль IDF? Какие ограничения имеет TF-IDF?
16. Что такое word embeddings?
17. В чём идея Word2Vec и GloVe?

**Список источников**

1. Учебное пособие «Основы искусственного интеллекта». Глава 9 «Обработка естественного языка (NLP)»: токенизация, векторизация, анализ тональности, машинный перевод.
2. Бессмертный И.А., Авдюшина А.Е., Кугаевских А.В., Королева Ю.А., Рущенко Н.Г. Системы искусственного интеллекта. Методическое пособие. Санкт-Петербург: Университет ИТМО, 2024.
3. Рабочая программа дисциплины «Искусственный интеллект и машинное обучение». Направление подготовки 09.03.01 «Информатика и вычислительная техника». Москва, 2024.
4. Сидоркина И.Г. Системы искусственного интеллекта. Учебное пособие. Москва: КНОРУС, 2015.
5. Бондарев В.Н., Аде Ф.Г. Искусственный интеллект. Учебное пособие. Севастополь, 2002.