16. Практическое занятие: Сокращение текста, суммаризация и архитектура Трансформеров

Цель занятия

Изучить методы автоматического сжатия текста (суммаризации), понять разницу между экстрактивным и абстрактивным подходами, а также познакомиться с архитектурой Transformer, которая лежит в основе современных языковых моделей.

Теоретический минимум

1. **Экстрактивная суммаризация:** выбор наиболее значимых предложений из исходного текста и их объединение. Текст не меняется, только сокращается.
2. **Абстрактивная суммаризация:** генерация нового текста, который передает смысл оригинала. Похоже на то, как человек пишет пересказ.
3. **Архитектура Transformer:** модель, основанная на механизме **Self-Attention** (самовнимание). Она позволяет модели фокусироваться на разных частях предложения одновременно, понимая контекст гораздо глубже, чем старые RNN-сети.

Задание 1. Экстрактивная суммаризация на основе TextRank

Алгоритм TextRank работает аналогично PageRank от Google: он строит граф предложений, где связи определяются семантическим сходством. Самые «важные» узлы графа становятся частью резюме.

Инструкция:

Используйте библиотеку gensim или простую реализацию на networkx для выделения ключевых предложений.

Python
```
from gensim.summarization import summarize

text = """
Искусственный интеллект — это область компьютерных наук, которая
занимается созданием систем, способных выполнять задачи, требующие
человеческого разума. К таким задачам относятся зрительное восприятие,
распознавание речи, принятие решений и перевод с одного языка на другой. В
последние годы ИИ совершил огромный скачок благодаря развитию глубокого
обучения и увеличению вычислительных мощностей. Современные модели, такие
как трансформеры, позволяют анализировать гигантские объемы данных и
генерировать тексты, которые трудно отличить от человеческих.
"""

# Сокращение текста до 50% объема
summary = summarize(text, ratio=0.5)
print("Экстрактивное резюме:\n", summary)
```

Задание 2. Абстрактивная суммаризация с использованием Hugging Face

Для генерации пересказов используются предобученные модели типа BART, T5 или GPT.

Инструкция:

Запустите предобученную модель для русского языка (например, IlyaGusev/mbart_ru_sum_gazeta) через интерфейс pipeline.

Python
```
from transformers import pipeline

# Загрузка пайплайна суммаризации (может потребоваться время на скачивание модели)
summarizer = pipeline("summarization", model="IlyaGusev/mbart_ru_sum_gazeta")

article = """
Ученые из МГУ разработали новый метод очистки океана от пластика с помощью специальных нанороботов.
Эти микроскопические устройства способны притягивать частицы микропластика и собирать их в крупные кластеры,
которые затем легко извлечь фильтрами. Эксперименты показали эффективность в 98%.
"""

summary_abstractive = summarizer(article, max_length=50, min_length=10, do_sample=False)
print("Абстрактивное резюме:", summary_abstractive[0]['summary_text'])
```

Задание 3. Исследование механизма Self-Attention

Ключевая особенность Трансформеров — матрица внимания. Она показывает, насколько одно слово в предложении «связано» с другим.

Инструкция:

Рассмотрите предложение: «Банк открыл новый филиал, потому что он расширяется».

1. На какое слово должен обратить «внимание» токен **«он»**? (Банк или филиал?).
2. Как механизм внимания помогает разрешить эту неоднозначность в архитектуре Трансформера?

Задание 4. Оценка качества: Метрика ROUGE

Качество суммаризации оценивается по метрике ROUGE (Recall-Oriented Understudy for Gisting Evaluation), которая сравнивает перекрытие слов в машинном резюме и эталонном (человеческом).

Инструкция:

Рассчитайте простое перекрытие (ROUGE-1) для двух фраз.

- Эталон: «Кошка сидит на ковре»
- Модель: «Кошка на ковре»
- $Recall = \frac{\text{Кол-во общих слов}}{\text{Кол-во слов в эталоне}} = 3/4 = 0.75$

Задание 5. Архитектурный анализ: Encoder vs Decoder

Трансформеры делятся на:

- **Only Encoder** (BERT) — идеальны для понимания текста и классификации.
- **Only Decoder** (GPT) — идеальны для генерации текста.
- **Encoder-Decoder** (T5, BART) — стандарт для суммаризации и перевода.

Контрольное задание:

Объясните, почему для задачи сокращения текста лучше всего подходит архитектура Encoder-Decoder? Что делает Encoder, а что — Decoder?

Контрольные вопросы

1. В чем главный недостаток экстрактивной суммаризации по сравнению с абстрактивной?
2. Почему Трансформеры вытеснили рекуррентные нейронные сети (RNN) в задачах обработки текста?
3. Что такое «галлюцинации» в абстрактивной суммаризации и как их минимизировать?

Итог работы

Вы изучили вершину современной обработки текста. Навыки суммаризации и понимание архитектуры Трансформеров позволяют создавать инструменты для быстрого анализа новостей, подготовки отчетов и работы с большими базами знаний.