ChatGPT

Модели и методы контекстно-ориентированного разбиения текста

Для RAG- и GraphRAG-конвейеров важно разбивать длинные тексты (транскрипты стримов) на смысловые фрагменты («чанки»), которые целиком отражают одну тему. Такое семантическое разбиение позволяет сохранять контекст и избегать произвольных границ. GraphRAG и другие RAG-системы рекомендуют, например, *семантическое (context-aware) чанкание*, при котором текст сначала разбивается на предложения, затем генерируются эмбеддинги, а разбиение проводится там, где расстояние между векторами предложений резко возрастает [1] [2]. Таким образом, для русского языка можно использовать комбинированный подход: сначала сегментировать текст на предложения, затем найти тематические границы по эмбеддингам смежных предложений.

В открытых репозиториях существуют готовые модели сегментации текста и инструменты контекстного чанкания:

- **Segment Any Text (SaT)** – универсальная модель (Token Classification) для сегментации предложений и абзацев. На Hugging Face представлены варианты SaT (однослойные и трёхслойные) «State-of-the-art sentence segmentation» с поддержкой 85 языков [3] [4]. Это лёгкие трансформеры (есть вариант на 1 слое, удобно запускать на CPU), обученные устойчиво работать без пунктуации (например, на неразмеченной стенограмме). Модель продаётся под MIT-лицензией, сообщество широко использует её для задачи chunking. Поддержка русского языка имеется в мультиязычных версиях (85 языков) [3]. SaT можно применить для наглядного разбиения транскриптов на осмысленные предложения и небольшие параграфы, сохраняя тему.

- **Segmentext (PleIAs)** – модель токен-классификации, обученная на разноформатных текстах (включая «грязные» OCR-данные). Segmentext умеет различать структурные элементы (заголовки, абзацы, диалоги, таблицы и т.д.) и «текстовые разделители» [5]. Она повышает качество чанкания за счёт понимания макроструктуры документа. Хотя ориентирована на европейские языки, её способность выделять, например, диалоговые блоки или списки может помочь и при транскриптах стримов, где часто меняются говорящие. Модель с открытым исходным кодом (Apache 2.0) и весами ~0.3B параметров [5]. Segmentext не рассчитана специально на русский, но её можно попробовать в сочетании с предобученными русскими токенизаторами (модели на HF легко адаптировать).

- **Docling (IBM)** – фреймворк для контекстно-ориентированного чанкания сложных документов. Docling разбирает PDF/Markdown на семантические элементы и создает «умные» чанки (учитывая заголовки, таблицы, списки и пр.) [6]. Так, в RHEL AI 1.3 добавлено «context aware chunking» на основе Docling: инструмент автоматически выделяет текстовые блоки, таблицы и изображения, создавая структурированные фрагменты [6]. Хотя это решение рассчитано на формальные документы, а не потоковую речь, его идея полезна для понимания: нужно

стараться разбивать текст по естественным границам (темы, спикеры, формат). Docling – open-source (GitHub IBM/Docling) и активно используется для сложных шаблонов.

- **Семантическое разбиение на основе эмбеддингов**. Как предлагает GraphRAG, можно вычислять эмбеддинги (например, российские Sentence-BERT или другие модели предложений [7]) и разрезать текст там, где косинусное расстояние между соседними векторами растёт. При этом сами модели эмбеддингов (например, *ai-forever/sbert_large_nlu_ru* [7] или *cointegrated/rubert-tiny2*) дают числовые представления смысловых блоков. Если два соседних предложения по смыслу слабо связаны, ставим границу чанка. Этот метод не требует обучения специфической модели для чанкания – достаточно предобученной модели эмбеддингов.

- **Простые алгоритмические подходы**. В качестве бенчмарка можно использовать классические методы (например, TextTiling из NLTK) или регулярное деление по предложениям (SpaCy, razdel) [8]. Однако они не учитывают глобальный контекст (что может вести к «рваному» чанканию при хаотичной речи). Тем не менее сочетание их с семантической проверкой (см. выше) часто улучшает результат.

Таким образом, для русского транскрипта стрима стоит попробовать гибрид: **SaT** (либо аналогичные BERT-модели сегментации) для выделения предложений и абзацев [3] [4], а затем разрывать чанки по смысловым скачкам (по эмбеддингам соседних предложений [2]). При этом инструмент **Docling** может помочь для структурированных источников (PDF), а **Segmentext** – для текста с неочевидной структурой. Все перечисленные решения – открытые: SaT и Segmentext доступны на Hugging Face, Docling – на GitHub IBM. Для встраивания на CPU подойдёт SaT в лёгкой конфигурации (1 слой), а также «тривиальные» методы NLTK/PySBD, дополненные встроенной проверкой семантической близости.

Источники: официальная документация GraphRAG по чанкованию [1] [2] и модельные репозитории Hugging Face (SaT [3] [4], Segmentext [5]), а также обзорный материал Red Hat по Docling [6].

---

[1] [2] [8] Text Chunking | GraphRAG
https://graphrag.com/guides/chunking/

[3] [4] segment-any-text/sat-1l · Hugging Face
https://huggingface.co/segment-any-text/sat-1l

[5] PleIAs/Segmentext · Hugging Face
https://huggingface.co/PleIAs/Segmentext

[6] RHEL AI 1.3 Docling context aware chunking: What you need to know
https://www.redhat.com/en/blog/rhel-13-docling-context-aware-chunking-what-you-need-know

[7] ai-forever/sbert_large_nlu_ru · Hugging Face
https://huggingface.co/ai-forever/sbert_large_nlu_ru