Tavily Summary Assignment

Yoav Shkedy, August 12, 2025

A. Research Orientation & Key Readings

I began with highly cited and recent references to ground design choices, metrics, and trade-offs:

- **Text Summarization Techniques: A Brief Survey** (arXiv:1707.02268)
  What I used: taxonomy of extractive vs. abstractive pipelines; limits of n-gram metrics (e.g., ROUGE), classic features (TF–IDF, graph centrality).
- **LexRank: Graph-based Lexical Centrality as Salience in Text Summarization** (arXiv:1109.2128)
  What I used: cosine/IDF-modified similarity graph, thresholding, PageRank for sentence salience. Mirrored key hyperparameters (e.g., threshold ≈ 0.1, damping as in paper).
- **Advancements in Natural Language Processing for Automatic Text Summarization** (arXiv:2502.19773)
  What I used: modern long-document strategies (hierarchical/recursive), latency-conscious design for production.
- **Multi-LLM Text Summarization**. (arXiv:2412.15487)
  What I used: Recursive compression as inspiration for our Advanced strategy.
- **A Comprehensive Survey on Automatic Text Summarization with Exploration of LLM-Based Methods** (arXiv:2403.02901)
  What I used: feasibility of fine-tuning/knowledge distillation (KD) for summarization and the operational implications (serving, context limits). I scoped FT/KD as a stretch due to time and hosting.

B. Approaches Tried

Lite (Extractive) — *LexRank on cleaned sentences*

- **Preprocessing:** boilerplate, links removal, noise filtering (`preprocess_lite.clean_web_text`).
- **Graph:** TF–IDF sentence vectors; cosine similarity; sparsify with threshold 0.1.
- **Ranking:** PageRank on row-stochastic matrix.
- **Selection:** Top sentences with redundancy control (Jaccard); optional original-order restoration; stop at `max_chars`.
- **Why LexRank (vs. TF–IDF+MMR/TextRank):** better global salience via centrality; stable under ROUGE-style overlap; robust to domain noise.

Balanced (Hybrid) — *LexRank evidence → small LLM rewrite*

- Run Lite to produce a rich extractive summary (3,000 chars).
- Feed that to a small LLM (**Amazon Nova Micro**) with a tuned prompt; produce a fluent summary $\leq$ **1200 characters**.
- *Why this works:* extractive step preserves coverage cheaply; the LLM fixes coherence/fluency with low latency and cost.

Advanced (Quality Ceiling) — *Recursive long-doc synthesis*

- Stage 1: **Recursive compression** with Amazon Nova Lite over overlapping chunks until under a token budget.
- Stage 2: **Final cohesive summary** with Claude Sonnet 4.
- *Why:* handles very long/complex pages; best readability and cohesion; highest cost/latency.

C. Experimentation Process

- **Paper-first scoping:** narrowed to extractive core + hybrid rewrite, reserving FT/KD as stretch.
- **Algorithm trials:** TF–IDF+MMR → TextRank → LexRank (kept). Matched LexRank paper settings.
- **Prompt/model sweeps:** several small LLMs; settled on **Nova Micro** for price/latency quality sweet spot (prompt refined for length, language, and style constraints).

D. Challenges & Limitations

- **Long, noisy HTML:** boilerplate and repeated nav/footer require aggressive cleaning to avoid extractive drift.
- **Multilingual segmentation:** non-Latin scripts and mixed-language pages make it complicate.
- **Extractive coherence:** sentence centrality can yield choppy discourse; LLM rewrite mitigates but cannot invent missing context.
- **Timeframe constraints:** FT/KD deemed stretch due to training, serving, and evaluation complexity; hosting custom models close to users adds DevOps overhead.
- **Evaluation fairness:** very long sources exceed context for LLM judges; we plan to use Lite summarizer for G-Eval reference document.