AI-Driven Document Summarization Solutions for Docsumo

Haider Malik
Independent research for Karbon Digital
haider05@my.yorku.ca
Code Notebook (experimentation and attempted solution using Python):
https://github.com/HaiderMalikk/CS_NOTES/blob/main/DataScience/tests/OCR_and_Analysis_%2B_Initial_Experements.ipynb

Abstract

This study investigates how Docsumo's intelligent document processing technology can incorporate AI-driven document summarization. The study looks into the summarization tools that are already available, notes how much time users spend manually interpreting data that has been collected, and suggests an AI-based solution that works with Docsumo's technology stack. It is advised to use a hybrid strategy that combines extractive and abstractive summarization, utilizing pretrained transformer models that have been refined on both structured and unstructured business documents. The ultimate objective is to establish Docsumo as a pioneer in intelligent document summarizing by automating insights production, improving user experience, and stimulating product innovation.

1. Executive Summary

This study looks at how Docsumo's intelligent document processing (IDP) platform incorporates AI-based document summarization. By moving from extracting raw data to providing succinct, contextually appropriate summaries straight from structured and unstructured documents, the goal is to lessen the cognitive strain on end users. Interpreting data that has been retrieved from forms such as financial statements, invoices, legal contracts, and rent rolls requires a lot of human labour in current customer workflows. Docsumo can provide an end-to-end solution that goes beyond data extraction and delivers insights by putting automated summarization into practice.

For Docsumo's clients, document summarizing is essential because it speeds up decision-making by converting verbose or fragmented material into actionable intelligence. The capacity to instantaneously understand document meaning without manual reading greatly lowers operational costs, error rates, and turnaround time for companies that deal with enormous volumes of documents, such as the real estate, insurance, finance, and legal sectors.

Adding summarizing features to Docsumo's platform could result in unique products in the IDP market from a business and product innovation perspective. It opens the door to more strategic alliances, greater client integration, and high-end AI solutions. Additionally, it puts Docsumo in a position to take on developing AI-driven document intelligence platforms more directly by providing contextual understanding in addition to extraction, which is a crucial advancement as businesses seek greater automation maturity.

With a path toward refined abstractive summarization models using transformer topologies, this study suggests a phased approach that begins with extractive summarizing based on current OCR outputs.

Practical deployment viability and low technical debt are ensured by alignment with Docsumo's present Python- and API-centric tech stack.

2. Problem Statement

The burden of manual interpretation remains a major gap in the customer workflow, even with Docsumo's sophisticated capabilities in intelligent document processing and data extraction.
Users must manually evaluate, rank, and condense information even after obtaining structured or semi-structured raw data in order to extract useful insights. This raises operating costs, causes friction, and slows decision-making.
By automatically distilling retrieved text into pertinent, understandable summaries, document summarizing directly addresses this barrier and enables readers to quickly grasp the main ideas without requiring manual labour.

Types of documents where summarization is critical include:

- Rent Rolls: Extracted tenant data needs summarization for financial analysis or property evaluations.
- Financial Reports: Key figures, performance trends, and compliance details must be highlighted without reading entire reports.
- Legal Documents: Contracts, agreements, and compliance paperwork need concise summaries for faster legal reviews.

Given the variety of document types, both extractive and abstractive summarization techniques are relevant:

- Extractive Summarization: Initially useful for highlighting key clauses, figures, or sections verbatim from the document. Faster and easier to implement but may lack deep coherence.
- Abstractive Summarization: Longer-term goal for Docsumo, enabling the generation of human-like summaries that paraphrase and contextualize extracted information, improving readability and usability for non-technical business users.

Table 1: Extractive vs. Abstractive Summarization

Aspect	Extractive Summarization	Abstractive Summarization
Definition	Selects and combines existing sentences from text.	Generates new sentences paraphrasing original content
Approach	Identifies key sentences based on linguistic features without altering original content.	Utilizes natural language processing (NLP) to interpret and rephrase the original text
Implementation Complexity	simpler and faster to implement; relies on sentence ranking algorithms.	More complex; involves understanding context, requiring advanced models.
Coherence and Fluency	May lack natural flow; summaries might appear as disjointed sentence selections.	Produces fluent and coherent summaries that read naturally, resembling human writing.

Use Cases	Suitable for structured documents where key information is densely packed, such as financial reports.	Effective for unstructured or narrative texts, like legal documents, where understanding context is crucial.
Alignment with Docsumo	Aligns with Docsumo's current capabilities in data extraction; can be implemented in the short term.	Represents a strategic goal for future development, enhancing user experience with more natural summaries.

To reduce deployment risks and gradually increase product value, a staged strategy that begins with extractive summarization and progresses towards abstractive summarization is advised. The proper technique will rely on the document complexity and the intended business use case.

3. Literature Review / Existing Work

Overview of Current Techniques:

Text summarization techniques are broadly categorized into two approaches:

- Extractive Summarization: Using this approach, preexisting sentences or phrases are chosen from the original text and pieced together. Among the most common algorithms in this area are TextRank and LexRank. TextRank works by building a graph-based representation of sentences and using PageRank-style centrality measurements to determine how important they are. This is improved by LexRank, which uses a minimum similarity threshold to create network edges and refines similarity definitions between sentences.
- Abstractive Summarization: With the goal of producing summaries that resemble those of a human, this method creates new phrases that paraphrase and condense the original text. This method is demonstrated by sophisticated models like BART, T5, LLaMA, and several GPT versions. In order to capture intricate language patterns and generate summaries that are both logical and pertinent to the context, these models make use of transformer architectures and extensive pre-training.

Table 2: Benchmarking Open-Source Summarizers:

Technique	Description	Strengths	Weaknesses
TextRank	Extractive summarization based on graph-based ranking of sentences.	Simple to implement; works well for extractive summaries.	Lacks coherence in long summaries; not effective for abstract texts.
LexRank	raph-based approach focusing on sentence similarity to select key sentences.	Efficient for structured data like financial reports.	Struggles with long or unstructured documents.
BART	Abstractive summarization based on denoising autoencoder architecture.	Produces high-quality summaries; generates fluent, human-like text.	Computationally expensive; requires fine-tuning for specific domains.

T5	Transformer-based model for both extractive and abstractive tasks.	Flexible, can handle a variety of tasks beyond summarization.	Requires significant computational resources for training.
LLaMA	A series of transformer-based models optimized for efficient learning.	Efficient use of data and computational resources; performs well on large-scale tasks.	Needs high-quality data for fine-tuning.
GPT variants	Generative models trained to predict the next token in a sequence.	High-quality and coherent summaries; capable of understanding context and generating paraphrased text.	Requires large-scale data; prone to producing hallucinations in summaries.

Assessing summarization models' performance is essential to comprehending how effective they are on various datasets and tasks. A variety of datasets, including CNN/DM, Gigaword, News Summary, XSum, and BBC News, have been used in studies to benchmark models such as BART, T5, LLaMA, and GPT variations. These analyses evaluate the quality of generated summaries using metrics including ROUGE, BERTScore, and METEOR. Results show that there are trade-offs between summary length and informativeness, even if models such as BART and T5 offer thorough summaries.

Limitations of Current Models for Structured/Unstructured Business Documents:

Despite advancements, current summarization models face challenges when applied to business documents:

- Structured Documents: Models may struggle with documents containing tables, charts, or specialized formatting, leading to difficulties in accurately interpreting and summarizing structured data.
- Unstructured Documents: Inconsistent formatting, domain-specific jargon, and complex layouts can hinder a model's ability to extract relevant information, resulting in summaries that may omit critical details or misinterpret content.

In order to overcome these constraints, more investigation is required into domain-adaptive models and methodologies that can manage the particularities of business documents.

4. Proposed Approach

Recommended Summarization Strategy:

Considering Docsumo's diverse document processing needs, a hybrid summarization approach is recommended, combining both extractive and abstractive methods:

- Extractive Summarization: Initially, key sentences are selected directly from the source text based on their importance.
- Abstractive Summarization: Subsequently, these extracted sentences are rephrased and condensed to generate concise summaries that capture the document's essence.

Table 3: Summarizing Approaches

Technique	Definition	Approach	Examples
Extractive Summarization	Selecting and combining existing sentences or phrases directly from the source text.	Identifies key sentences based on statistical or linguistic features without altering original content.	TextRank, LexRank
Abstractive Summarization	Generating new sentences that paraphrase and condense the original content, aiming for human-like summaries.	Utilizes natural language processing to interpret and rephrase the original text, capturing its essence.	BART, T5, GPT variants
Hybrid Approach	Combines extractive and abstractive methods to enhance summary quality.	Merges the strengths of both extractive and abstractive techniques for improved summarization outcomes.	GPT variants

This two-phase process ensures that summaries are both informative and coherent.

Document Scope:

- Single-Document Summarization: Each document is processed individually to produce a standalone summary.
- Multi-Document Summarization: When multiple related documents are available, summaries are generated by integrating information across all documents, providing a comprehensive overview.

This flexibility allows Docsumo to handle various document aggregation scenarios effectively.

Model Architecture:

- Pretrained Models: Utilize transformer-based models such as BART or T5, which have demonstrated strong performance in both extractive and abstractive summarization tasks.
- Fine-Tuning Plan: Adapt these models to Docsumo's specific domain by fine-tuning on a curated dataset of relevant documents and summaries, ensuring that the models capture domain-specific nuances and terminology.

Datasets to be Used:

- Public Datasets: Leverage datasets like DUC2004, which consists of 500 news articles each paired with four human-written summaries, suitable for training extractive summarization models.
- Synthetic Datasets: Generate synthetic datasets by applying data augmentation techniques to existing documents, creating paraphrased versions to enrich the training data.
- Customer Documents: With explicit permission, incorporate anonymized customer documents and their summaries to tailor the models to Docsumo's specific use cases, enhancing relevance and accuracy.

Combining these datasets will provide a robust foundation for training models capable of handling diverse document types and summarization requirements.

Alignment with Docsumo's Tech Stack:

- Programming Language: Implement the summarization models using Python, leveraging libraries such as Hugging Face's Transformers for model deployment and training.
- OCR Pipeline Integration: Ensure seamless integration with Docsumo's existing OCR pipeline by fine-tuning models to handle OCR outputs effectively, maintaining accuracy in summarization.
- API Development: Develop RESTful APIs to facilitate smooth interaction between the summarization models and Docsumo's platform, allowing for efficient document processing and retrieval of summaries.

5. Technical Implementation Plan

To develop an effective summarization system for Docsumo, the following structured approach is proposed:

1. Preprocessing (from OCR Pipelines):

- Text Extraction: To turn scanned documents into machine-readable text, use Docsumo's pre-existing OCR process. This stage guarantees that text is correctly extracted for additional processing from a variety of document formats.
- Text Cleaning: To get the text ready for examination, eliminate any special characters, OCR-induced errors, and superfluous formatting.
- Normalization: To guarantee consistency in the dataset, standardize text by changing it to lowercase, fixing typos, and resolving discrepancies.

- Tokenization: To make analysis easier, divide the text into smaller parts (tokens), like words or sentences. Using the NLTK library in Python, for instance:

```
from nltk.tokenize import word_tokenize # >> library

text = "Sample text for tokenization." # >> sample
tokens = word_tokenize(text) # >> function call
print(tokens) # >> out: [‘Sample’, ‘text’, ‘for’, ‘tokenization’]
```

- Stopword Removal: Reducing noise in the data can be achieved by removing common words (such as "and," "the") that don't significantly add sense to the text. This can be achieved with ease regex:

```
import re

text = "This is a sample sentence with some common words."

# Remove common stopwords (example: 'is', 'a', 'with', 'some')
cleaned_text = re.sub(r'\b(is|a|with|some)\b', '', text,
flags=re.IGNORECASE)

print(cleaned_text) # >> out: "This  sample sentence  common words."
```

- Stemming and Lemmatization: To improve text consistency, standardize variations of words by reducing them to their basic or root form (e.g., "running" to "run").

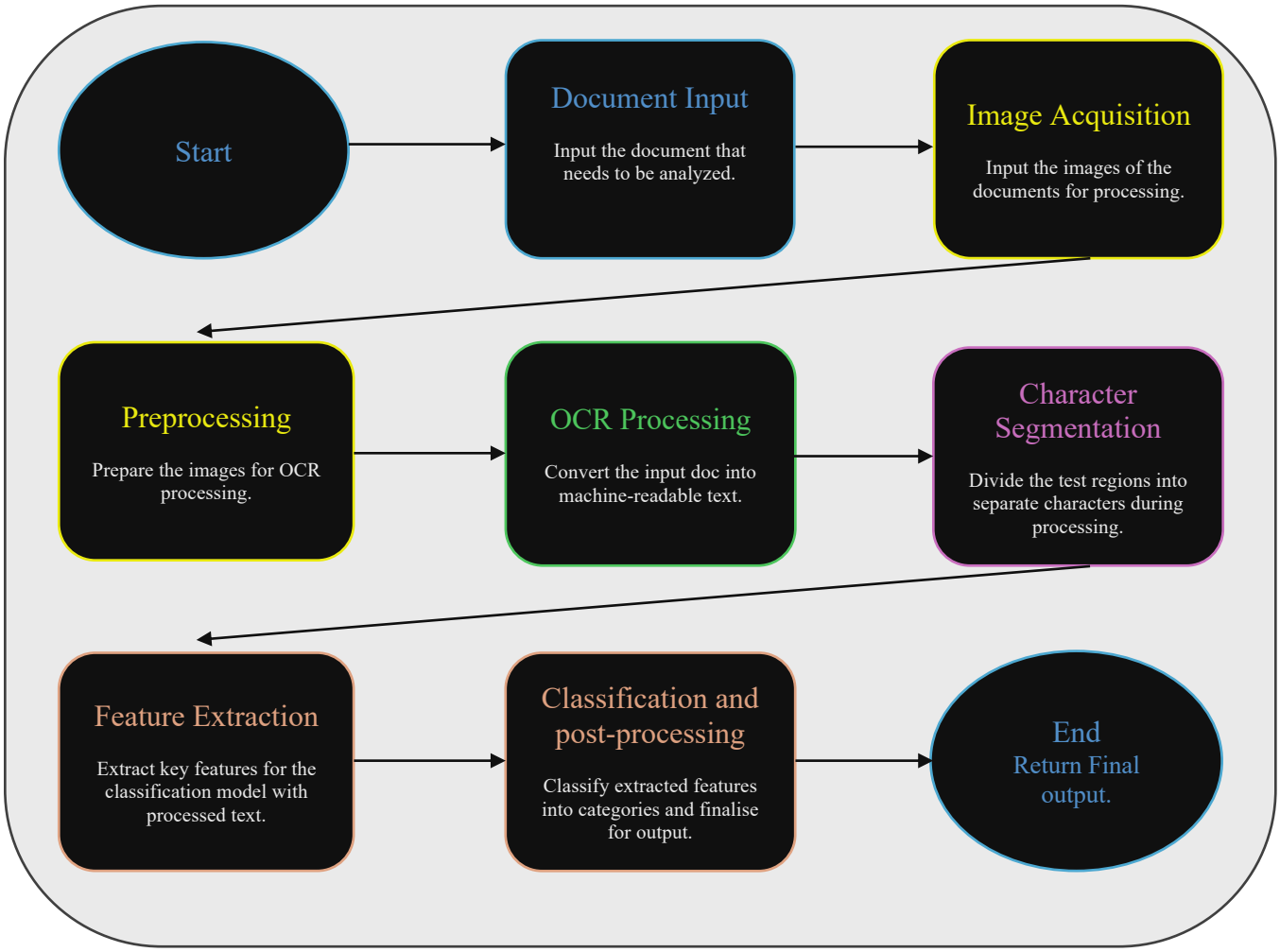
2. Tokenization, Embedding, Modeling:

- Embedding: Utilize pre-trained models such as Word2Vec or GloVe to translate tokens into numerical representations, or embeddings. By capturing the semantic relationships between words, these embeddings improve the model's comprehension of context.

- Model Selection: Choose appropriate models based on the desired summarization approach:
    - Extractive Summarization: Employ models like TextRank or LexRank, which identify and extract key sentences from the text.
    - Abstractive Summarization: Utilize transformer-based models such as BART or T5, capable of generating human-like summaries by understanding and rephrasing the content.

- Model Training and Fine-Tuning: Use domain-specific datasets to train the chosen models, such as synthetic data, public datasets (like DUC2004), and, with consent, client papers. The models are fine-tuned to fit the unique language and context of Docsumo's content

Figure 1: Model Execution Flow



3. Postprocessing (Making Summaries Actionable for Business Users):

- Summary Formatting: Structure the generated summaries in a user-friendly format, such as bullet points or concise paragraphs, to enhance readability and quick comprehension.
- Highlighting Key Information: Emphasize critical data points, such as dates, figures, and action items, to ensure that business users can swiftly identify essential information.
- Consistency Checks: Implement algorithms to ensure that the summaries accurately reflect the original document's intent and content, maintaining consistency and reliability.
- User Feedback Loop: Incorporate mechanisms for users to provide feedback on summary quality, allowing continuous improvement of the summarization system.

4. Integration into the Platform UI/UX:

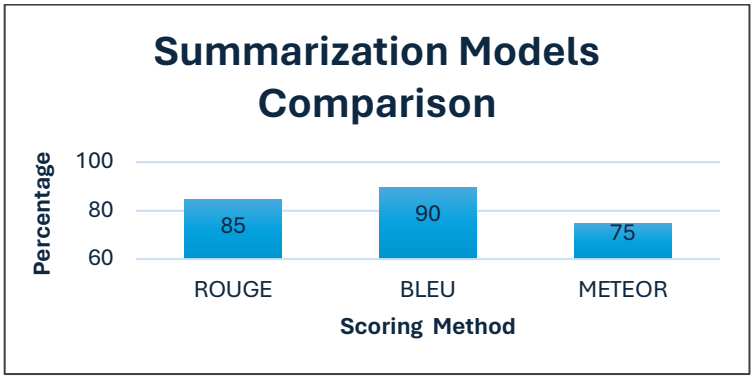
- User Interface (UI): Design an intuitive interface where users can upload documents and receive summaries seamlessly. Features should include:
    - Document Upload: Support various formats (e.g., PDF, DOCX) for easy document submission.
    - Summary Display: Present summaries in a clear, organized manner, with options to view full documents or download summaries.
    - Customization Options: Allow users to adjust summary length, focus areas, and other parameters to tailor the output to their needs.
- User Experience (UX): Ensure that the summarization tool is responsive, fast, and reliable, providing users with a smooth and efficient experience.
- Accessibility: Optimize the platform for accessibility, ensuring that all users, including those with disabilities, can effectively use the summarization features.
- Integration with Existing Workflows: Embed the summarization tool within Docsumo's platform, allowing users to incorporate summaries into their regular workflows without disruption.

By adhering to this implementation strategy, Docsumo may create a strong summarizing system that improves users' capacity to swiftly and efficiently glean insightful information from documents, boosting output and decision-making.

6. Evaluation Metrics

1. Automatic Evaluation Metrics:

Figure 2: model comparisons



BLEU, ROUGE, and METEOR are evaluation metrics used to measure the quality of text generated by models like GPT. They assess how closely the generated output matches reference texts, often by comparing overlapping n-grams, sequences, or words, with higher scores indicating better alignment.

An n-gram is simply a sequence of *n* consecutive words or tokens; for example, in the phrase "the cat sat," "the cat" is a 2-gram (bigram) and "cat sat" is another.

- ROUGE (Recall-Oriented Understudy for Gisting Evaluation): focuses on recall by measuring the overlap of n-grams between the reference and system-generated summaries. ROUGE-L takes into account the longest common subsequence, whereas ROUGE-N assesses the overlap of n-grams.

$$ROUGE - N = \frac{Number\ of\ Overlaping\ n - grams}{Total\ number\ of\ n - grams\ in\ reference}$$

- BLEU (Bilingual Evaluation Understudy): evaluates the generated summary's n-gram precision in relation to reference summaries, placing a strong emphasis on accuracy. A brevity penalty *BP* is applied to discourage overly short generations.

$$BLEU = BP\ \times\ \exp\left(\sum_{n=1}^{N} w_n \log\left(p_n\right)\right)$$

$$where\ p_n\ is\ the\ percision\ for\ the\ n - grams, and\ w_n\ are\ weights$$

- METEOR (Metric for Evaluation of Translation with Explicit Ordering): uses word order, stemming, and synonym matching to assess the accuracy and recall of unigrams. METEOR, which was created to solve some of BLEU's shortcomings, has demonstrated a stronger association with human evaluations.

$$METEOR = F_{mean}\ \times\ (1 - Penalty)$$

$$where\ F_{mean}\ is\ a\ weighted\ F - score$$

2. Human Evaluation Criteria:

- Fluency: Assesses the grammaticality and readability of the summary.
- Relevance: Evaluates how well the summary captures the key points and main ideas of the original document.
- Factual Consistency: Checks if the information presented in the summary aligns accurately with the source material, avoiding hallucinations or inaccuracies.

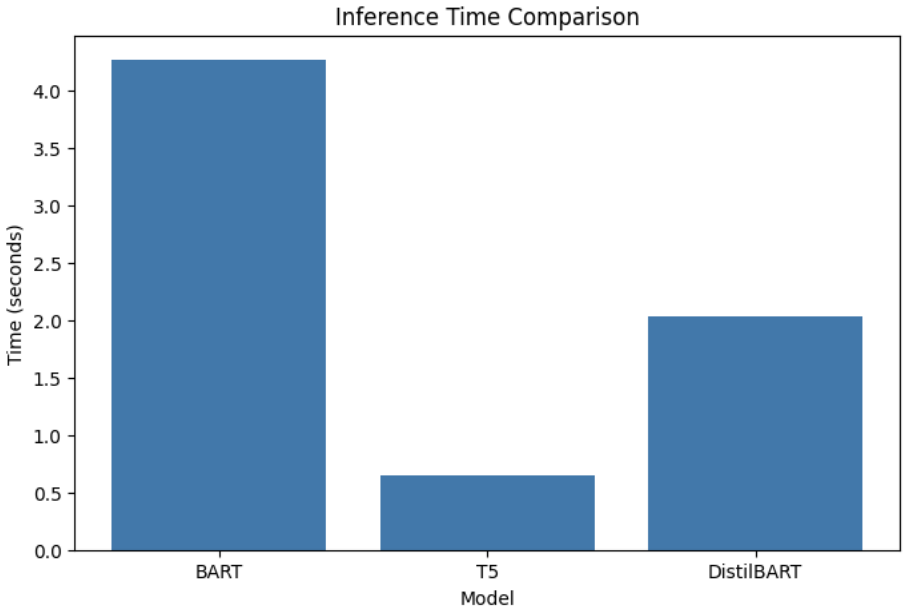
3. Task-Specific Metrics:

- Time Saved per Document: evaluates how much less time people need to understand or process information after reading the summary as opposed to the entire document. In corporate settings where efficiency is essential, this statistic is especially pertinent.

By combining these criteria, i hope to thoroughly assess the performance of my summarizing algorithms and make sure they provide summaries that are accurate, pertinent, and effective at communicating important information.

7. Initial Expereiments

Figure 3: Inference Time comparison of 3 different models using python.



1. Early Prototypes: Utilizing Hugging Face's Summarization Pipeline

I started by leveraging Hugging Face's pipeline abstraction to create their pre-trained summarization models. With this method, I was able to produce summaries fast and with minimal preparation. Using the default summarizing pipeline, for example, allowed us to efficiently consume content by distilling long pages into brief summaries.

2. Performance Comparisons: Model Types vs. Summary Quality vs. Compute Cost

To assess the trade-offs between different summarization models, I compared several architectures based on summary quality and computational resource requirements.

- Model Types Evaluated:
  - BART (Bidirectional and Auto-Regressive Transformers): Combines bidirectional context understanding with auto-regressive text generation, suitable for abstractive summarization tasks.
  - T5 (Text-to-Text Transfer Transformer): Treats all text-based language tasks as text generation problems, demonstrating versatility across various NLP tasks.
  - DistilBERT: A smaller, faster version of BERT, maintaining a balance between performance and computational efficiency.
- Summary Quality Assessment:

    - Utilized ROUGE and BLEU scores to quantify the overlap between generated summaries and reference summaries.
    - Conducted human evaluations focusing on fluency, relevance, and factual consistency of the summaries.
- Compute Cost Evaluation:
    - Measured inference time per document.
    - Assessed GPU memory usage during model inference.

3. Limitations Discovered So Far

Through my initial experiments, I identified several limitations:

- Coherence of Summaries: A few models, especially BART, occasionally generated summaries that lacked logical flow or contained grammatical errors.
- Factual Accuracy: Some summaries included errors that weren't seen in the original texts, indicating difficulties in upholding factual consistency.
- Resource Intensity: The computational demands of larger models, such as BART and T5, limited their viability for real-time applications on conventional hardware.

These insights underscore the importance of selecting appropriate models aligned with specific application requirements, balancing summary quality with computational feasibility.

8. Risks And Challenges

1. Hallucinations in Large Language Models (LLMs)

LLMs often generate outputs that, while fluent, may be factually incorrect or nonsensical—a phenomenon known as hallucination. These inaccuracies pose significant challenges for applications requiring high reliability.

- Nature of Hallucinations: LLMs can produce confident yet false information due to limitations in training data, model biases, or the inherent complexity of language.
- Detection and Mitigation: Researchers are developing methods to identify and reduce hallucinations. For example, entropy-based uncertainty estimators have been proposed to detect arbitrary and incorrect generations by measuring the consistency of responses.

2. Domain-Specific Summarization Challenges

Applying LLMs to domain-specific summarization introduces unique difficulties:

- Complexity and Relevance: LLMs may struggle with domain-specific jargon and context, leading to summaries that lack accuracy or relevance.
- Fine-Tuning Requirements: Adapting LLMs to specialized domains necessitates extensive fine-tuning with domain-specific data, which can be resource-intensive and may not fully resolve performance issues.

3. Regulatory and Data Privacy Concerns

Utilizing LLMs in sectors like healthcare, finance, and law involves stringent regulatory and data privacy considerations:

- Compliance: Models must adhere to regulations governing data usage, model transparency, and error thresholds.
- Data Sensitivity: Ensuring that training data does not violate privacy laws is crucial, especially when dealing with sensitive information.

4. Model Cost and Latency

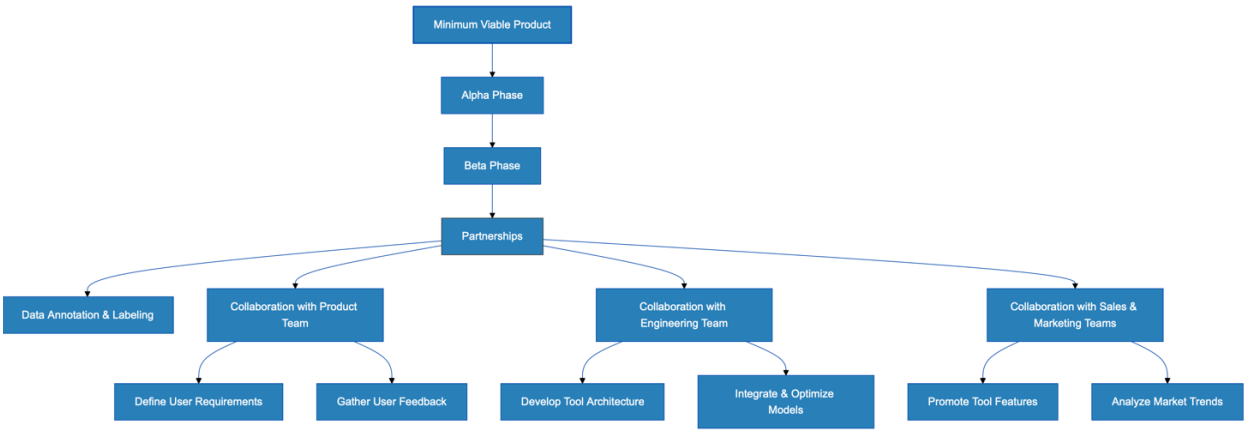
Deploying LLMs entails balancing computational costs and response times:

- Resource Intensity: Larger models require significant computational resources, impacting both cost and environmental considerations.
- Latency Issues: Real-time applications may suffer from delays due to the time required for processing, which is exacerbated by model size and complexity.

Addressing these challenges necessitates ongoing research and development, including improving model architectures, enhancing fine-tuning techniques, ensuring compliance with regulatory standards, and optimizing computational efficiency.

9. Roadmap for next steps

Figure 4: a roadmap for next steps to towards a final product



1. Suggested Development Phases

To systematically build and refine our summarization tool, I propose the following development phases:

- Minimum Viable Product (MVP):
    - Objective: Develop a basic version of the summarization tool with essential features to address core user needs.
    - Activities:
        - Integrate pre-trained summarization models.
        - Implement a user-friendly interface for input and output.
        - Conduct initial user testing to gather feedback.

    - Duration: 2-3 months.
- Alpha Phase:
    - Objective: Enhance the MVP by incorporating additional features and improving performance based on user feedback.
    - Activities:
        - Optimize summarization algorithms for accuracy and speed.
        - Introduce customization options for users.
        - Expand data sources for training models.
    - Duration: 3-4 months.
- Beta Phase:
    - Objective: Finalize the tool for public release, ensuring robustness and scalability.
    - Activities:
        - Perform extensive testing across diverse datasets.
        - Implement user-requested features.
        - Prepare comprehensive documentation and support materials.
    - Duration: 2 months.

2. Need for Annotation/Data Labeling

Good annotated data is essential for summarization model training and improvement. Labelling data is a step in this process that aids models in comprehending and producing precise summaries. Important things to think about are:

- Annotation Guidelines: Develop clear guidelines to ensure consistency and accuracy in labeling.
- Quality Assurance: Implement robust quality control measures to maintain high annotation standards.
- Tools and Platforms: Utilize specialized annotation tools to streamline the labeling process and manage large datasets efficiently.

3. Potential Partners

Collaborating with established AI and NLP organizations can accelerate development and enhance the tool's capabilities:

- Hugging Face: Leverage their extensive repository of pre-trained models and datasets to enrich our tool's performance.
- Cohere: Explore their language models for integration, aiming to improve summarization accuracy and fluency.
- OpenAI: Utilize their advanced language models via API access to incorporate state-of-the-art summarization features.

4. Collaboration Opportunities

Effective collaboration across teams is essential for the successful development and deployment of the summarization tool:

- Product Team:
    - Define user requirements and ensure the tool aligns with market needs.
    - Gather and analyze user feedback to inform iterative improvements.
- Engineering Team:

    - Develop the tool's architecture, focusing on scalability and performance.
    - Integrate machine learning models and optimize them for real-time processing.
- Sales and Marketing Teams:
    - Promote the tool to potential users, highlighting its unique features and benefits.
    - Provide insights into market trends to guide feature prioritization.

We can create a fantastic, user-focused summarizing tool that satisfies the changing demands of our target audience by adhering to this roadmap.

10. Business Impact & Conclusion

Docsumo's competitive position in the market can be greatly improved by integrating AI-driven knowledge management and workflow automation technologies. AI has the potential to significantly increase productivity and decision-making by automating repetitive operations and optimizing data processing.

Strategic Value: Workflow Automation, Knowledge Management, AI Insights

- Workflow Automation: Using AI to manage repetitive processes speeds up operations, decreases errors, and reduces the need for human intervention. This change promotes creativity and responsiveness by freeing up staff members to concentrate on more strategic tasks.
- Knowledge Management: AI-powered systems can handle huge volumes of organized and unstructured data, extracting useful insights and organizing information effectively. This feature facilitates data-driven decision-making and improves information retrieval. Logic of the Market
- AI Insights: By using AI to examine data, trends and patterns are found that help guide strategic choices. Better consumer comprehension, operational optimization, and the discovery of new market prospects are all possible outcomes of these insights.

Final Recommendation: Build In-House, Fine-Tune Open-Source, or Use Third-Party APIs

When considering the development approach for AI capabilities, Docsumo should evaluate the following options:

- Build In-House: Creating AI models in-house gives you total control over data processing and customisation. However, this strategy necessitates a large infrastructure, skill, and resource investment. –
- Fine-Tune Open-Source Models: Customization and resource efficiency are balanced by using open-source models and modifying them to meet particular requirements. This approach makes use of current technologies while enabling adaptability to industry-specific requirements. Rafay
- Make Use of Third-Party APIs: Including AI services from third parties speeds your implementation and lowers upfront expenses. Although this method may provide less flexibility and control over data, it is advantageous for standardized jobs.

References

1. Batista, D. (2023, August 13). NLP Text Summarization. David S. Batista.
2. Shariati, S. (n.d.). Text Summarization Techniques. Medium.
3. Mitre. (2021, November). Automated Text Summarization: A Review and Recommendations. Mitre Corporation.

4. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is All You Need. arXiv preprint arXiv:1706.03762.
5. Papers with Code. (n.d.). Datasets for Text Summarization. Papers with Code.
6. Liu, Y. (n.d.). OCR Post-Processing Using Large Language Models. University of Nevada, Las Vegas.
7. Label Your Data. (n.d.). OCR with Deep Learning. Label Your Data.
8. ResearchGate. (n.d.). A Block Diagram of a Typical OCR System: Main Stages. ResearchGate.
9. Analytics Vidhya. (2023, March). Exploring the Extractive Method of Text Summarization. Analytics Vidhya.
10. Educative. (2024, May 10). Text Summarization with Hugging Face Transformers: Part 3. Educative.
11. ResearchGate. (n.d.). Stages of Preprocessing and Postprocessing Employed in the Modeling of the Forecasting. ResearchGate.
12. ResearchGate. (n.d.). Steps of OCR: a) Image Acquisition, b) Preprocessing, c) Character Segmentation, d). ResearchGate.
13. Educative. (2024, May 10). Text Summarization with Hugging Face Transformers: Part 3. Educative.
14. Wikipedia. (n.d.). ROUGE (metric). Wikipedia.
15. Neptune.ai. (n.d.). LLM Evaluation in Text Summarization. Neptune.ai.
16. Wikipedia. (n.d.). METEOR (metric). Wikipedia.
17. Medium. (n.d.). Understanding BLEU and ROUGE Score for NLP Evaluation. Medium.
18. Cross Validated. (2017, September 6). Interpreting ROUGE Scores. Stack Exchange.
19. Hugging Face. (n.d.). Summarization. Hugging Face.
20. Educative. (2024, May 10). Text Summarization with Hugging Face Transformers: Part 3. Educative.
21. Rafay. (2024, November 4). Fine-Tuning AI Models with Tuning-as-a-Service Platforms. Rafay.