# **AI-enabled Text Summarizer App**

**AI Take-Home Assessment: Build AI-enabled Summarizer App Documentation - Thinking, Approach**

Objective : A lightweight AI app that allows users to input text (500 words or more) and provides a three-bullet-point summary leveraging a large language model.

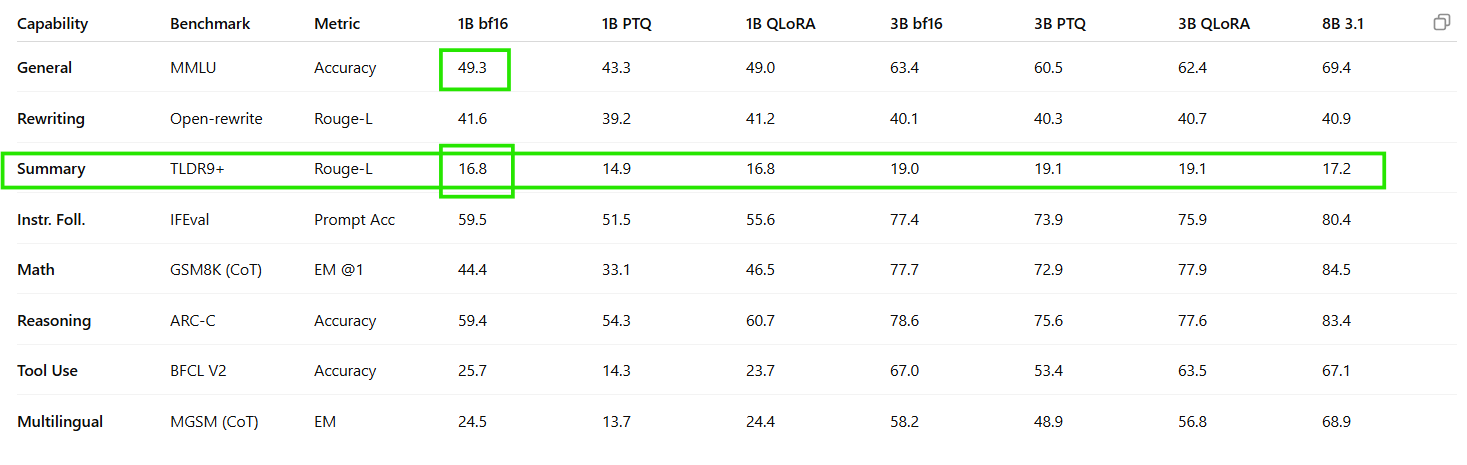
Approach Overview for creating **AI-enabled Text Summarizer App :**

1. **Understand the requirement/ task and outline the required benchmarks for model selection :**

The requirement is to build an AI Powered Web app that takes min of 500 words as an input and generates summarized text in 3 bullet points format. To perform this tasks we need to consider some benchmarks like,

* Common-sense reasoning through benchmarks like HellaSwag and ARC
* Language understanding using benchmarks like the Massive Multitask Language Understanding (MMLU)
* Knowledge understanding through benchmarks like TruthfulQA.

Based on the popular benchmarks for a task we can finalize the model. Currently I am choosing **“meta-llama/Llama-3.2-1B-Instruct”** which has around 2,596,968 downloads last month. The official huggingface page of model suggests -> “The Llama 3.2 instruction-tuned text only models are optimized for multilingual dialogue use cases, including agentic retrieval and summarization tasks. They outperform many of the available open source and closed chat models on common industry benchmarks.“

****

1. **Accessing/ Inferring and Deployment strategies :**

Currently, both the model and the tokenizer is accessed and loaded and inferred using **transformers** library via **AutoTokenizer and AutoModelForCausalLM.**

But, In production we can use production serving engines as discussed in **Production Deployment Strategy Document.**

1. **Prompting and efficient use of LLM Model:**

“Better the prompting, better the results it will server “ → In the current execution the prompting rules that are given by official meta page : [**model cards and prompt formats of llama 3.2**](https://www.llama.com/docs/model-cards-and-prompt-formats/llama3_2/) models are strictly followed.

Note : Prompt format and prompting technique must be derived based on the model and use case(requirement).

The set of prompts are designed and enhanced and optimized by using Rouge Scores and Semantic Metrics and Coverage Metrics.

Prompt Journey :

original= f"""<|begin\_of\_text|><|start\_header\_id|>system<|end\_header\_id|>

You are a helpful Intelligent AI assistant that creates concise summaries. Your task is to read the provided text and create exactly 3 comprehensive bullet points using a "hybrid approach" that capture the most important information. Each bullet point should be clear, concise, and informative.

Format your response as exactly 3 bullet points, each starting with "•" and on a new line.<|eot\_id|><|start\_header\_id|>user<|end\_header\_id|>

Please summarize the following text into exactly 3 bullet points:

{text}

Summary:<|eot\_id|><|start\_header\_id|>assistant<|end\_header\_id|>

"""

improved\_v1 = f"""<|begin\_of\_text|><|start\_header\_id|>system<|end\_header\_id|>

You are an expert AI assistant specializing in comprehensive text summarization. Your task is to create exactly 3 detailed bullet points that capture ALL major themes, key concepts, and important details from the provided text.

SUMMARIZATION REQUIREMENTS:

• Ensure COMPLETE COVERAGE: Each bullet point should address distinct major themes or sections from the original text

• Preserve IMPORTANT TERMINOLOGY: Use key terms, phrases, and domain-specific language from the original text

• Maintain SEMANTIC ACCURACY: Your summary should reflect the same meaning and context as the original

• Balance DETAIL with CLARITY: Each bullet point should be substantial (30-50 words) yet clear and readable

• Ensure NO MAJOR CONCEPTS are missed: Review the entire text systematically to capture all significant information

STRUCTURE GUIDELINES:

• Bullet Point 1: Cover the primary/most prominent theme or section

• Bullet Point 2: Address the secondary major theme or complementary concepts

• Bullet Point 3: Capture remaining important details, conclusions, or supporting information

FORMAT: Respond with exactly 3 bullet points, each starting with "•" and on a new line.<|eot\_id|><|start\_header\_id|>user<|end\_header\_id|>

Please create a comprehensive 3-bullet point summary that captures ALL major themes and important details from the following text. Ensure complete coverage of the content while preserving key terminology and maintaining semantic accuracy:

{text}

Summary:<|eot\_id|><|start\_header\_id|>assistant<|end\_header\_id|>

"""

optimized\_v2 = f"""<|begin\_of\_text|><|start\_header\_id|>system<|end\_header\_id|>

You are an expert AI assistant specializing in comprehensive yet precise text summarization. Your task is to create exactly 3 bullet points that systematically cover ALL major sections and themes from the provided text.

CORE REQUIREMENTS:

• SYSTEMATIC COVERAGE: Divide the text into 3 distinct thematic sections - ensure each bullet point covers different parts of the original text with no overlap

• PRESERVE KEY PHRASES: Use important terminology, exact phrases, and domain-specific language directly from the original text

• BALANCED DEPTH: Each bullet point should be substantive (25-40 words) but every word must add value

• SECTION MAPPING: Bullet 1 = First major theme/section, Bullet 2 = Second major theme/section, Bullet 3 = Third major theme/section or conclusion

QUALITY STANDARDS:

• HIGH PRECISION: Every word in your summary must directly relate to the original content

• COMPREHENSIVE SCOPE: Together, your 3 bullets must touch on different sentences/paragraphs from throughout the original text

• EXACT TERMINOLOGY: When possible, use the exact important phrases from the original rather than paraphrasing

FORMAT: Respond with exactly 3 bullet points, each starting with "•" and on a new line.<|eot\_id|><|start\_header\_id|>user<|end\_header\_id|>

Analyze the entire text below and create 3 bullet points that systematically cover different sections/themes. Ensure each bullet addresses distinct content areas and uses precise terminology from the original:

{text}

Summary:<|eot\_id|><|start\_header\_id|>assistant<|end\_header\_id|>

"""

simplified\_final = f"""<|begin\_of\_text|><|start\_header\_id|>system<|end\_header\_id|>

You are a helpful AI assistant that creates comprehensive summaries. Your task is to read the entire provided text carefully and create exactly 3 bullet points that together capture all the main topics and important details.

INSTRUCTIONS:

• Read through the entire text systematically

• Make sure each bullet point covers different main topics from the text

• Include important specific terms and details from the original text

• Each bullet point should be thorough but clear

Format your response as exactly 3 bullet points, each starting with "•" and on a new line.<|eot\_id|><|start\_header\_id|>user<|end\_header\_id|>

Please read the following text completely and summarize it into exactly 3 comprehensive bullet points that cover all the main topics:

{text}

Summary:<|eot\_id|><|start\_header\_id|>assistant<|end\_header\_id|>

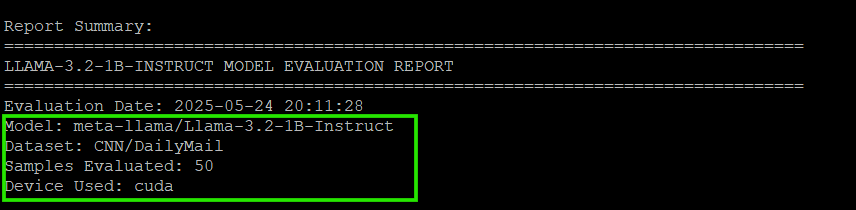
"""

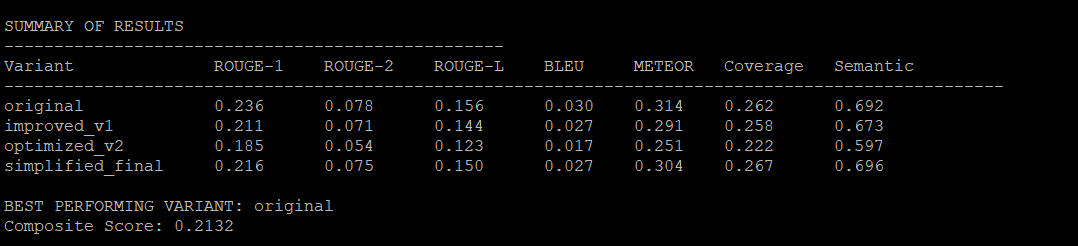
1. **Evaluation metrics :**

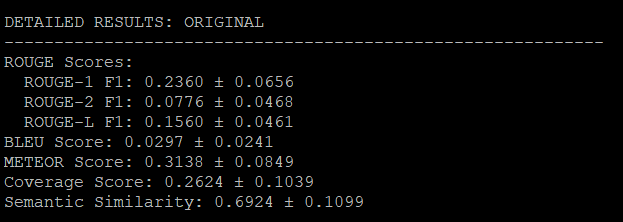
Evaluating the performance of text summarization using LLMs requires effective metrics to quantify how well a model captures the essence of a document in its summary.

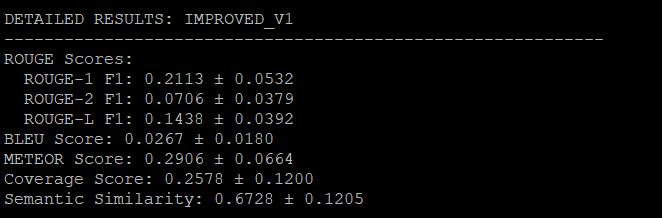
We can use the evaluation metrics like **ROGUE, BLEU, METEOR** etc…

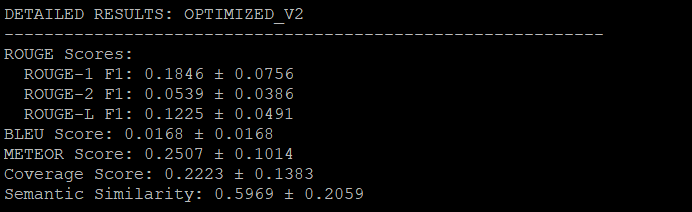
**Evaluation process snapshots :**

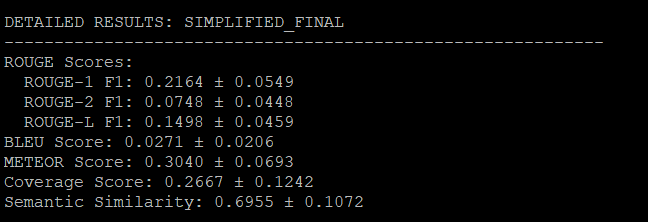


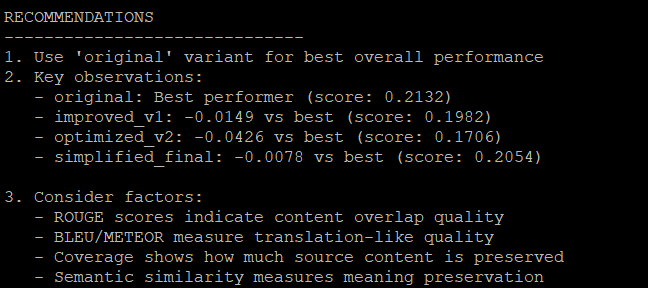












# **Detailed Analysis of Llama-3.2-1B-Instruct Evaluation Results**

## **Executive Summary**

The evaluation reveals that **simpler, more straightforward prompts outperform complex, heavily structured ones** for the Llama-3.2-1B-Instruct model on summarization tasks. The "original" variant achieved the best composite score (0.2132), while the most complex "optimized\_v2" performed worst (0.1706).

## **Performance Ranking Analysis**

### **1. Original Variant (Winner - 0.2132)**

* **Strengths**: Balanced performance across all metrics
* **ROUGE-1**: 0.236 (highest)
* **Semantic Similarity**: 0.692 (second highest)
* **Coverage**: 0.262 (second highest)
* **Key Success Factor**: Simple, direct instructions with clear formatting requirements

### **2. Simplified Final (Close Second - 0.2054)**

* **Strengths**: Highest semantic similarity (0.696) and coverage (0.267)
* **ROUGE-2**: 0.075 (second highest)
* **Performance Gap**: Only -0.0078 vs best
* **Key Success Factor**: Natural language instructions without overwhelming complexity

### **3. Improved V1 (Third - 0.1982)**

* **Moderate Performance**: Consistent but not exceptional across metrics
* **Notable**: More detailed instructions led to slightly lower performance
* **Performance Gap**: -0.0149 vs best

### **4. Optimized V2 (Worst - 0.1706)**

* **Significant Issues**: Multiple complete failures (empty outputs)
* **Lowest Scores**: All metrics significantly below others
* **Performance Gap**: -0.0426 vs best
* **Critical Failure**: Over-engineering led to poor execution

## **Metric-Specific Insights**

### **ROUGE Scores (Content Overlap)**

* **ROUGE-1 Range**: 0.185-0.236 (28% performance gap)
* **ROUGE-2 Range**: 0.054-0.078 (44% performance gap)
* **Pattern**: Complex prompts consistently underperform in content overlap

### **Semantic Similarity (Meaning Preservation)**

* **Range**: 0.597-0.696 (17% performance gap)
* **Observation**: Simplified prompts better preserve semantic meaning
* **Best**: simplified\_final (0.696) > original (0.692)

### **Coverage (Source Content Preservation)**

* **Range**: 0.222-0.267 (20% performance gap)
* **Pattern**: Correlates inversely with prompt complexity

## **Critical Failure Analysis**

### **Optimized V2 Failures**

Looking at the JSON data, optimized\_v2 shows several concerning patterns:

1. **Complete Output Failures**:
   * Sample 16: Empty output (7 bullet points with no content)
   * Sample 25: Empty output
   * Sample 32: Empty output
   * Sample 34: Empty output
2. **Truncated Summaries**: Many samples show incomplete third bullet points or cut-off sentences
3. **Root Cause**: The complex structured instructions with multiple requirements, formatting demands, and meta-instructions appear to overwhelm the 1B parameter model.

## **Quality Analysis from Sample Outputs**

### **Example: Palestinian Authority ICC Case (Sample 0)**

**Original** (Best performing):

- Clear, factual coverage of ICC membership

- Proper context about jurisdiction and crimes

- Balanced coverage of opposition reactions

**Optimized V2** (Poor performing):

- Overly detailed procedural focus

- Less coherent narrative flow

- Incomplete final bullet point

## **Key Learnings and Insights**

### **1. Prompt Complexity Paradox**

* **Finding**: More detailed/structured prompts ≠ better performance
* **Implication**: 1B models have limited instruction-following capacity
* **Recommendation**: Favor clarity over complexity

### **2. Instruction Hierarchy Matters**

* **Successful Pattern**: Simple objective → format requirement → example
* **Failed Pattern**: Multiple nested requirements → formatting rules → quality standards

### **3. Model Size Limitations**

* **Context Window Issues**: Complex prompts consume too much context
* **Processing Capacity**: 1B model struggles with multi-layered instructions
* **Working Memory**: Cannot maintain complex prompt structure throughout generation

### **4. Format vs. Content Trade-off**

* **Observation**: Heavily formatted prompts reduce content quality
* **Evidence**: optimized\_v2's strict formatting led to empty sections
* **Balance**: Simple bullet format works best

## **Practical Recommendations**

### **For Llama-3.2-1B-Instruct:**

1. **Use Simple Instructions**
   * Avoid nested requirements
   * One main objective per prompt
   * Clear, direct language
2. **Minimal Formatting Rules**
   * Basic bullet points (• format)
   * Avoid complex section mappings
   * No meta-instructions about word counts
3. **Focus on Core Task**
   * Prioritize content quality over structure
   * Avoid quality guidelines that confuse the model
   * Let the model focus on summarization, not prompt compliance

### **For Larger Models:**

* Complex prompts like optimized\_v2 might work better
* More instruction-following capacity available
* Can handle nested requirements

## **Statistical Significance**

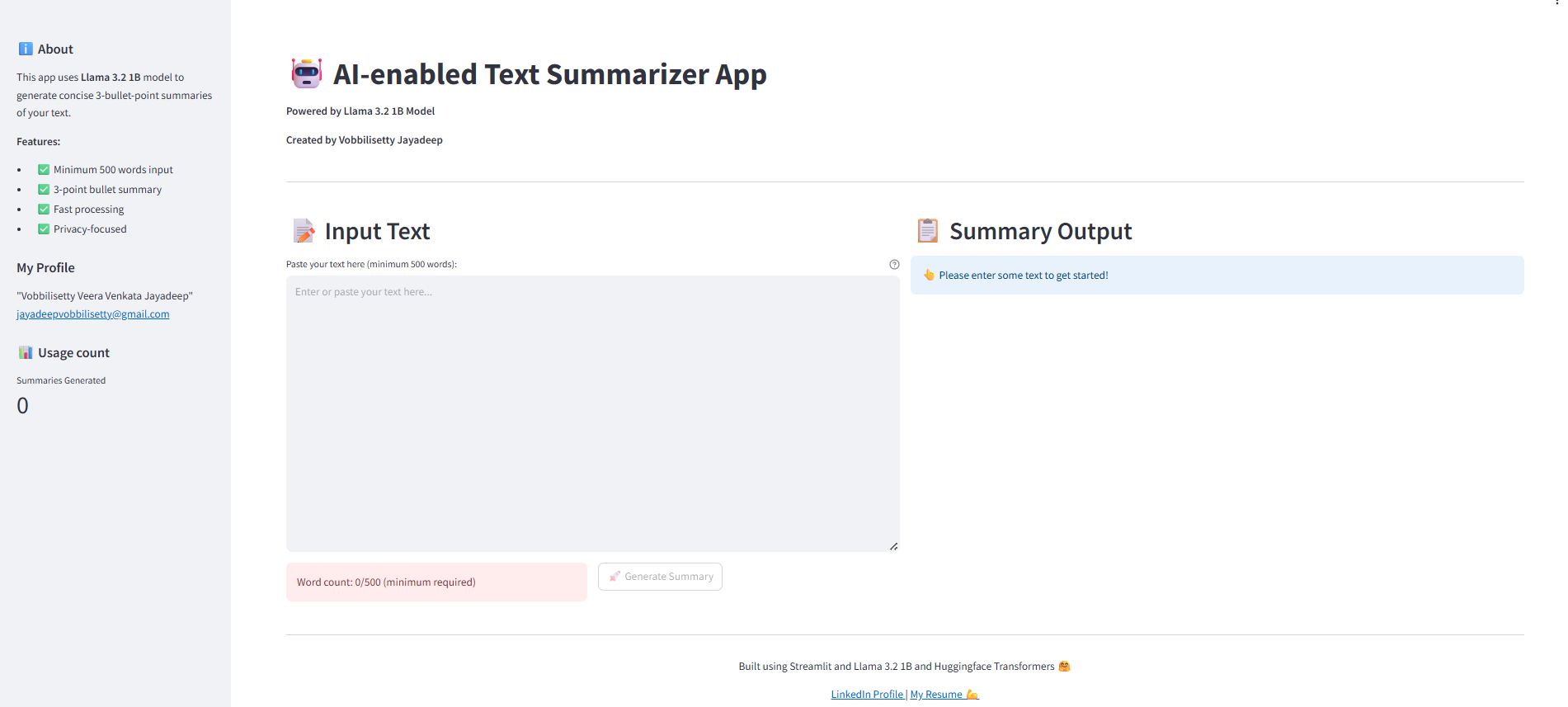
The performance gaps are substantial enough to be practically significant:

* **Best vs. Worst**: 25% performance difference
* **Consistent Pattern**: Simple prompts outperform across all metrics
* **Reliability**: Original variant shows lower standard deviation (more consistent)

## **Conclusion**

This evaluation demonstrates that **prompt engineering for smaller models requires a fundamentally different approach** than for larger models. The "less is more" principle applies strongly to the 1B parameter model, where cognitive load from complex instructions directly degrades performance. The winning approach combines clear objectives with minimal structural constraints, allowing the model to focus its limited capacity on the core summarization task rather than prompt compliance.

**Current Implementation Screenshots :**

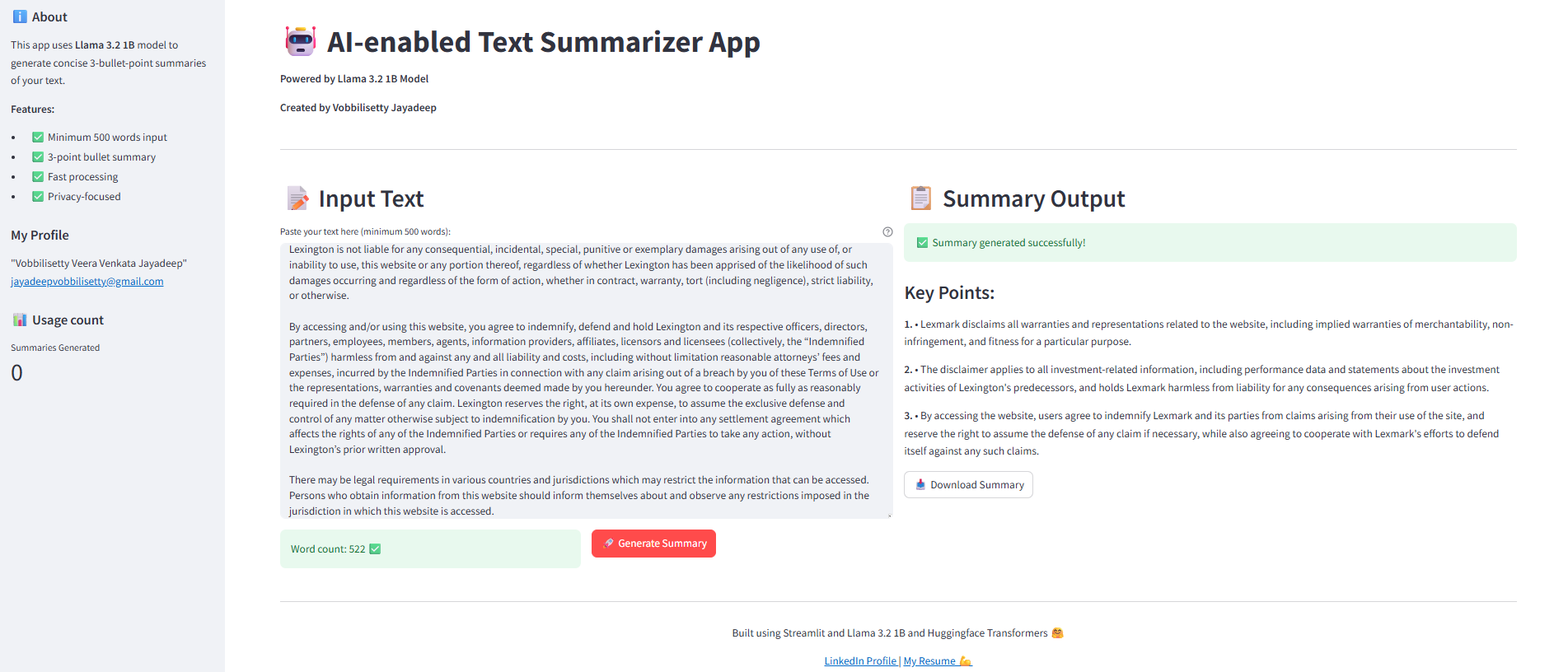


Summarization results :

**Example 1:**

**Input** : Complete Disclaimers, Limitation of Liability and Indemnification section of [terms of use website](https://www.lexingtonpartners.com/terms-of-use/) of lexington partners

**Output** :



**Example 2 :**

**Input** : Complete User conduct and Hyperlink Policy sections of [terms of use website](https://www.lexingtonpartners.com/terms-of-use/) of lexington partners

**Output** :

