Report: Tokenization in the Context of Large Language Models (LLMs)

Prepared by: Anbhi Thakur
Date: July 19, 2025

---

1. What is Tokenization in NLP and LLMs?

Tokenization is the process of converting raw text into smaller units called **tokens**. In the context of Natural Language Processing (NLP) and Large Language Models (LLMs), tokens are the atomic units of input — these can be:

- Words (e.g., "artificial", "intelligence")
- Subwords (e.g., "inter", "esting")
- Characters (e.g., "a", "b", "c")
- Byte-pair encodings (BPEs) like "intelli" + "gence"

Modern LLMs such as GPT, PaLM, LLaMA, and Claude use **subword-level tokenization**, often using BPE or **SentencePiece** to handle vocabulary efficiently across languages and formats.

Example: The sentence “ChatGPT is amazing!”
May be tokenized as: ["Chat", "G", "PT", " is", " amazing", "!"]
→ 6 tokens

---

2. Why Tokenization is Essential Before Feeding Text to LLMs

Tokenization is **foundational** for LLMs because:

a. Model Input Representation

LLMs operate on numbers, not raw text. Tokenization converts text into token IDs (integers), enabling the model to process and understand text using learned embeddings.

b. Language Generalization

By breaking words into subwords, tokenization allows LLMs to:

- Handle out-of-vocabulary words
- Better generalize across morphological variants (e.g., "run", "running", "runner")

c. Efficiency and Robustness

Tokenization enables more compact and flexible representations, improving performance on:

- Rare words
- Multilingual content

- Code and symbols

---

3. How Token Limits Impact Prompt Size, Inference Latency, and API Costs

LLMs have **token limits** that constrain how much data (input + output) they can handle in one request. These limits directly affect usage in several ways:

a. Prompt Size Constraints

- If a model has a 8K token limit, and 7K is used for input, only 1K remains for output.
- Longer prompts may need trimming or summarization before use.

b. Inference Latency

- More tokens = longer computation = higher latency.
- Token-by-token generation increases with token count linearly or more, depending on the model size and architecture.

c. API Cost Impact

- **OpenAI, Anthropic, etc., charge per 1,000 tokens** processed (input + output).
- Example: Processing 2,000 input + 1,000 output tokens at $0.03/1K = $0.09
- Reducing prompt size and tuning output length helps reduce cost.

Model (example)	Token Limit	Typical Cost (per 1K tokens)
GPT-4-turbo	128K	$0.01 input / $0.03 output
Claude 3 Opus	200K	$0.015 input / $0.075 output

Summary

- **Tokenization** breaks text into units (tokens) LLMs can understand.
- It is **essential** for converting raw text into model-digestible numerical data.
- Token limits directly impact **prompt capacity**, **speed**, and **API usage cost**.
- Understanding tokenization helps developers write **efficient**, **cost-effective**, and **high-performing** prompts.