Text Data Characteristics

Sequential Nature

Word order and context matter significantly

High Dimensionality

Vocabulary size can reach tens of thousands

Sparse Representation

Most words don't appear in any given document

Ambiguity

Words have multiple meanings depending on context

Language-Dependent

Different languages have unique structures and rules

Rich Semantic Info

Information encoded in syntax and grammar

Required Processing Steps

- Tokenization
- Normalization
- Vocabulary Management

Text Data Examples

Example 1: Product Review

```
"This laptop is AMAZING!!! Best purchase ever 😊"
```

Example 2: Social Media Post

```
"Can't believe it's already 2024... time flies! #newyear"
```

Example 3: Customer Inquiry

```
"Hi, I'm wondering if this product comes in blue?"
```

🔧 Processing Steps Example

1 Tokenization

Split text into individual tokens (words, subwords, or characters)

```
Input: "This laptop is AMAZING!!!"
Output: ["This", "laptop", "is", "AMAZING", "!", "!", "!"]
```

2 Normalization

Convert to lowercase, remove punctuation, handle special characters

```
Input: ["This", "laptop", "is", "AMAZING", "!", "!", "!"]
Output: ["this", "laptop", "is", "amazing"]
```

3 Vocabulary Management

Map tokens to unique IDs based on vocabulary dictionary

```
Input: ["this", "laptop", "is", "amazing"]
Vocabulary: {this: 45, laptop: 1203, is: 12, amazing: 789}
Output: [45, 1203, 12, 789]
```