

Tokenization (Tách Từ) - Nền Tảng Xử Lý Ngôn Ngữ Tự Nhiên (NLP)

Tokenization là gì?

Tokenization là quá trình chia nhỏ văn bản thành các đơn vị nhỏ hơn như từ, cụm từ hoặc câu, gọi là **token**.

Ví dụ:

“VHTSoft is a technology company”

→ **Tokens:** ["VHTSoft", "is", "a", "technology", "company"]

Tưởng tượng tokenization giống như việc **cắt một cuốn sách thành từng từ riêng lẻ hoặc tách các câu**—đây là bước cực kỳ quan trọng trong NLP và Hệ thống Truy vấn Thông tin (IR).

Tại sao Tokenization quan trọng?

1. Đơn giản hóa xử lý văn bản

- Biến dữ liệu thô thành các phần nhỏ, dễ phân tích.

2. Hỗ trợ đánh chỉ mục (indexing)

- Giúp tìm kiếm và truy xuất thông tin nhanh hơn (vd: search engine).

3. Làm sạch dữ liệu

- Loại bỏ stopwords (từ không quan trọng như "a", "the"), chuẩn hóa văn bản.

4. Giúp AI hiểu ngữ nghĩa

- Là đầu vào cho các mô hình NLP như BERT, GPT.

1. Word Tokenization (Tách từ)

- Chia văn bản thành các từ riêng lẻ.

- Ví dụ:

"Học máy rất thú vị" → ["Học", "máy", "rất", "thú", "vị"]

2. Sentence Tokenization (Tách câu)

- Chia văn bản thành các câu.
- Ví dụ:

“Tôi thích AI. Tôi học NLP.” → ["Tôi thích AI.", "Tôi học NLP."]

3. Character Tokenization (Tách ký tự)

- Chia văn bản thành từng ký tự.
- Ví dụ:

“AI” → ["A", "I"]

Thách Thức Khi Tokenization

- **Xử lý dấu câu:** Dấu chấm, phẩy có nên là token riêng?
- **Từ ghép:** "ice-cream", "mother-in-law"—nên tách hay giữ nguyên?
- **Ký tự đặc biệt:** Emoji, hashtag (#AI), URL.
- **Đa ngôn ngữ:**
 - Tiếng Việt: "Xin chào" → ["Xin", "chào"]
 - Tiếng Anh: "Hello" → ["Hello"]
 - Tiếng Nhật: *"[カタリ]"` (không có khoảng cách giữa từ).

Triển Khai Tokenization Trong Python

Sử dụng thư viện **NLTK** (Natural Language Toolkit):

```
import nltk
nltk.download('punkt') # Tải dữ liệu tokenizer

# Word Tokenization
from nltk.tokenize import word_tokenize
text = "VHTSoft is a technology company"
tokens = word_tokenize(text)
print("Word Tokens:", tokens) # Output: ['VHTSoft', 'is', 'a', 'technology', 'company']

# Sentence Tokenization
```

```
from nltk.tokenize import sent_tokenize
text = "I love AI. I study NLP."
sentences = sent_tokenize(text)
print("Sentence Tokens:", sentences) # Output: ['I love AI.', 'I study NLP.']
```

Tiền Xử Lý Văn Bản(TextPreprocessing)

Sau khi **tokenization**, bước tiếp theo là **chuẩn hóa văn bản** bằng cách:

1. **Chuyển thành chữ thường** (lowercase)
2. **Loại bỏ các token không phải chữ và số** (non-alphanumeric)

Triển Khai Preprocessing Trong Python

```
import re

from nltk.tokenize import word_tokenize


def preprocess(text):

    # 1. Chuyển thành chữ thường
    text = text.lower()

    # 2. Tokenization
    tokens = word_tokenize(text)

    # 3. Loại bỏ token không phải chữ/số (giữ lại từ có dấu)
    tokens = [token for token in tokens if re.match(r'^[a-z0-9àáâãäåæçèéêëẽëöóôõööðóðóðóðóðóðóðíîïúûüýÿđ]+$', token)]

    return tokens


# Ví dụ các tài liệu
documents = [

    "VHTSoft is a TECHNOLOGY company!",

    "AI, Machine Learning & NLP are COOL.",

    "Xử lý ngôn ngữ tự nhiên (NLP) rất quan trọng!"

]


# Tiền xử lý từng tài liệu
preprocessed_docs = [' '.join(preprocess(doc)) for doc in documents]


# Kết quả
```

```
for i, doc in enumerate(preprocessed_docs):  
    print(f"Document {i+1}: {doc}")
```

Kết Quả Sẽ Là:

Document 1: vhtsoft is a technology company
Document 2: ai machine learning nlp are cool
Document 3: xử lý ngôn ngữ tự nhiên nlp rất quan trọng

Giải Thích:

- `text.lower()`: Chuyển tất cả thành chữ thường để đồng nhất hóa.
- `re.match()`: Chỉ giữ lại token chứa chữ cái (kể cả tiếng Việt), số.
- `' '.join()`: Ghép các token lại thành câu sau khi xử lý.

Mẹo Thực Tế Để Tokenization Hiệu Quả

Dù đơn giản, **tokenization** là bước cực kỳ quan trọng để phân tích dữ liệu văn bản và tạo nền tảng cho các tác vụ NLP nâng cao. Dưới đây là những lời khuyên thiết thực:

1. Tiền Xử Lý Văn Bản (Preprocess Text)

- **Chuẩn hóa dữ liệu** trước khi tokenize:
 - Chuyển thành chữ thường (`lowercase`).
 - Loại bỏ ký tự đặc biệt (như `!?, @`), nhưng giữ lại từ có dấu (tiếng Việt).
 - Xử lý viết tắt (vd: "ko" → "không").
- **Ví dụ:**

```
text = "Tokenization là BƯỚC ĐẦU!!!"  
text = text.lower() # "tokenization là bước đầu!!!"
```

2. Xử Lý từ không mang nhiều ý nghĩa(Stopwords) (Handle Stopwords)

- **Stopwords** là những từ ít mang nghĩa (vd: "và", "là", "the").
- Nên loại bỏ chúng để giảm nhiễu, nhưng **cẩn thận với ngữ cảnh**:
 - Tiếng Việt: "**không** tốt" → Nếu xóa "không", nghĩa đảo ngược!
- **Cách làm:**
- **Tùy chỉnh danh sách stopwords để giữ lại từ như:**

```
stopwords = [...] # danh sách từ dừng mặc định  
important_words = ['không', 'chưa', 'chẳng', 'chả', 'đừng']
```

```
# Loại bỏ các từ phủ định khỏi stopwords
```

```
stopwords = [word for word in stopwords if word not in important_words]
```

- **Dùng mô hình học sâu hiểu ngữ cảnh (BERT tiếng Việt)**

Mô hình như **PhoBERT**, **viBERT**, hoặc **VietAI-BERT** đã được huấn luyện để **hiểu từ phủ định** theo ngữ cảnh, không cần xử lý thủ công.

Khi Nào Dùng Tokenization?

- Xây dựng chatbot, search engine.
- Phân tích cảm xúc (sentiment analysis).
- Xử lý dữ liệu trước khi đưa vào AI model.

Tác giả: Đỗ Ngọc Tú

Công Ty Phần Mềm VHTSoft

Phiên bản #1

Được tạo 23 tháng 4 2025 03:28:09 bởi Đỗ Ngọc Tú

Được cập nhật 23 tháng 4 2025 04:02:26 bởi Đỗ Ngọc Tú