**tạo vector từ văn bản** (còn gọi là vector hóa) là bước nền tảng và quan trọng nhất trong Xử lý Ngôn ngữ Tự nhiên (NLP). Mục tiêu là chuyển đổi văn bản, thứ mà con người hiểu, thành các con số (vector), thứ mà máy tính có thể xử lý và tính toán.

Dưới đây là các phương pháp chính để thực hiện việc này, từ cổ điển đến hiện đại.

**1. Các phương pháp dựa trên tần suất (Count-based)**

Các phương pháp này tạo vector bằng cách đếm số lần xuất hiện của từ. Chúng đơn giản nhưng hiệu quả cho nhiều bài toán cơ bản.

**Bag-of-Words (BoW - Túi từ)**

Đây là phương pháp đơn giản nhất.

* **Cách hoạt động:**
  1. Xây dựng một "từ điển" chứa tất cả các từ duy nhất trong toàn bộ kho văn bản của bạn.
  2. Với mỗi câu hoặc đoạn văn, tạo một vector có độ dài bằng kích thước từ điển.
  3. Tại mỗi vị trí trong vector, điền số lần từ tương ứng trong từ điển xuất hiện trong câu đó.
* **Ví dụ:**
  1. Từ điển: ["tôi", "thích", "phở", "bún"]
  2. Câu 1: "tôi thích phở" → Vector: [1, 1, 1, 0]
  3. Câu 2: "tôi thích bún" → Vector: [1, 1, 0, 1]
* **Nhược điểm:** Không phân biệt được thứ tự từ và không biết từ nào quan trọng hơn.

**TF-IDF (Term Frequency - Inverse Document Frequency)**

Đây là một cải tiến của BoW, giúp đánh giá mức độ quan trọng của một từ.

* **Cách hoạt động:**
  + **TF (Tần suất từ):** Tương tự BoW, đếm số lần từ xuất hiện trong một văn bản.
  + **IDF (Tần suất tài liệu nghịch):** Đánh giá mức độ "hiếm" của một từ. Những từ xuất hiện trong ít tài liệu (như "phở") sẽ có điểm IDF cao hơn những từ xuất hiện ở mọi nơi (như "là", "và").
  + Vector cuối cùng được tạo ra bằng cách nhân TF với IDF.
* **Kết quả:** Các từ vừa xuất hiện nhiều trong một văn bản cụ thể, vừa hiếm trong toàn bộ kho dữ liệu sẽ có trọng số cao nhất.

**2. Các phương pháp hiện đại (Embedding)**

Đây là phương pháp tiên tiến, tạo ra các **vector dày đặc (dense vector)** có khả năng nắm bắt được **ngữ nghĩa** và **ngữ cảnh** của từ/câu.

**Word Embedding (Word2Vec, GloVe)**

* **Cách hoạt động:** Các mô hình này học vector cho từng từ bằng cách phân tích mối quan hệ của nó với các từ xung quanh trong một kho văn bản khổng lồ.
* **Đặc điểm:** Các từ có nghĩa tương tự sẽ có vector gần nhau trong không gian. Điều này tạo ra các mối quan hệ toán học thú vị, ví dụ: vector("Vua") - vector("Nam") + vector("Nữ") ≈ vector("Nữ hoàng")

**Sentence Embedding (Sentence-Transformers)**

Đây là phương pháp mạnh mẽ và dễ sử dụng nhất hiện nay để tạo vector cho cả câu hoặc đoạn văn.

* **Cách hoạt động:** Sử dụng các mô hình Transformer đã được huấn luyện trước (pre-trained) trên lượng dữ liệu khổng lồ. Các mô hình này có khả năng hiểu sâu sắc về ngữ pháp, ngữ nghĩa và ngữ cảnh.
* **Ưu điểm:** Vector tạo ra không chỉ dựa vào các từ riêng lẻ mà còn cả cấu trúc và ý nghĩa của cả câu, cho kết quả vượt trội trong hầu hết các tác vụ.

**3. Code Python thực tế**

Cách dễ nhất để bắt đầu là sử dụng thư viện sentence-transformers. Dưới đây là đoạn mã để tạo vector chất lượng cao cho văn bản tiếng Việt.

**Tóm tắt**

* Để bắt đầu, hãy sử dụng phương pháp **Sentence Embedding** với thư viện sentence-transformers.
* Nó cung cấp các vector chất lượng cao, nắm bắt được ngữ nghĩa sâu sắc của văn bản.
* Các vector này là đầu vào lý tưởng cho các tác vụ NLP như **tìm kiếm ngữ nghĩa, phân loại văn bản, gợi ý nội dung, và gom cụm**.