Xây dựng Không Gian Vector Từ Tiếng Việt Sử dụng Mô hình Word2Vec và Tập Dữ liệu Lớn

Bùi Anh Chiến - 23021490

October 13, 2025

Abstract

Tóm tắt này trình bày quá trình xây dựng một mô hình nhúng từ (Word Embedding) Word2Vec cho tiếng Việt. Dự án sử dụng tập dữ liệu lớn và quy trình tiền xử lý mạnh mẽ, bao gồm phân tích từ ghép và lọc stopword dựa trên cấu trúc Trie. Mô hình được huấn luyện bằng kiến trúc Skip-Gram với kích thước vector là $D = 300$. Các kết quả định tính cho thấy mô hình học được mối quan hệ ngữ nghĩa và cú pháp hiệu quả, đặc biệt trong các bài kiểm tra tìm từ tương đồng và phân biệt từ khác loại (Odd One Out). Báo cáo cung cấp chi tiết về các siêu tham số, quy trình huấn luyện song song và đánh giá chất lượng mô hình.

1 Giới thiệu

Trong lĩnh vực Xử lý Ngôn ngữ Tự nhiên (NLP), biểu diễn từ ngữ thành các vector số học (Word Embeddings) đóng vai trò nền tảng. Các vector này, điển hình là Word2Vec [**?**], giúp máy tính hiểu được ngữ nghĩa và mối quan hệ giữa các từ. Các ngôn ngữ có cấu trúc phức tạp như tiếng Việt, với đặc điểm từ ghép và sự phụ thuộc ngữ cảnh cao, đòi hỏi quy trình tiền xử lý cẩn thận để đạt được chất lượng vector nhúng tối ưu.

Nghiên cứu này tập trung vào việc huấn luyện mô hình Word2Vec (Skip-Gram) trên một tập dữ liệu tiếng Việt lớn, sử dụng thư viện `underthesea` cho việc tách từ và xử lý từ ghép. Mục tiêu là tạo ra một không gian vector từ tiếng Việt chất lượng cao, có khả năng nắm bắt được cả mối quan hệ ngữ nghĩa và phép toán vector (Word Analogy).

2 Phương pháp luận

2.1 Dữ liệu huấn luyện (Corpus)

Tập dữ liệu được sử dụng là **VTSNLP/vietnamese_curated_dataset** [3], một corpus tiếng Việt lớn, công khai. Để xử lý khối lượng dữ liệu khổng lồ này một cách an toàn và hiệu quả, quy trình đã áp dụng cơ chế **Streaming** từ thư viện `datasets` của Hugging Face. Việc xử lý được thực hiện song song (Multiprocess) trên $N = 8$ tiến trình (dựa trên cấu hình `os.cpu_count() - 2` hoặc 4), với kích thước batch là 15.000 dòng.

2.2 Quy trình tiền xử lý

Tiền xử lý là bước quan trọng nhất đối với tiếng Việt. Các bước thực hiện bao gồm:

1. **Chuẩn hóa:** Chuyển đổi Unicode sang định dạng NFC (Normalization Form C) và đưa về chữ thường (`lower()`).

2. **Loại bỏ ký tự không hợp lệ:** Loại bỏ các ký tự không phải chữ cái tiếng Việt, chữ cái La-tinh cơ bản và số (`_re_non_vn`).

3. **Tách từ và Từ ghép:** Sử dụng thư viện `underthesea` với tham số `format="text"` để thực hiện phân đoạn từ (Word Segmentation) và tự động nối các từ ghép bằng dấu gạch dưới (`_`), ví dụ: "nhà_khoa_học".

4. **Lọc Stopword Nâng cao:** Áp dụng danh sách stopword ngoại lai (`stopwords.csv`). Để xử lý các cụm stopword đa từ, một cấu trúc **Trie** đã được xây dựng. Thuật toán lọc stopword sử dụng Trie cho phép nhận diện và loại bỏ các cụm từ (ví dụ: "bởi_vì_thế_nên") trong một lần quét duy nhất, tối ưu hóa hiệu suất.

2.3 Kiến trúc mô hình và Siêu tham số

Mô hình nhúng từ được huấn luyện là **Word2Vec** từ thư viện `gensim`. Các tham số chính được thiết lập như sau:

Table 1: Siêu tham số mô hình Word2Vec

Siêu tham số	**Giá trị**
Kiến trúc (`sg`)	Skip-Gram (1)
Kích thước Vector (D)	300
Kích thước Window (W)	5
Số Epochs (E)	15
Ngưỡng Tần suất (`min_count`)	5
Lấy mẫu âm (`negative`)	10
Ngưỡng Downsample (`sample`)	10^{-5}
Số Worker (`workers`)	14 (giới hạn phần cứng)

Giải thích lựa chọn siêu tham số:

- **Kiến trúc Skip-Gram:** Thích hợp khi muốn học các từ hiếm và mối quan hệ ngữ nghĩa phức tạp, vì Skip-Gram dự đoán các từ xung quanh từ trung tâm.
- **Kích thước vector ($D = 300$):** Kích thước 300 đủ để biểu diễn thông tin ngữ nghĩa mà không gây quá tải bộ nhớ.
- **Kích thước window ($W = 5$):** Cân bằng giữa học ngữ cảnh gần và ngữ cảnh rộng.
- **Số epochs ($E = 15$):** Đảm bảo mô hình được huấn luyện đầy đủ mà không gây overfitting.
- **Ngưỡng tần suất (`min_count`=5):** Loại bỏ các từ quá hiếm, giúp giảm noise.
- **Lấy mẫu âm (`negative`=10):** Giúp mô hình phân biệt tốt hơn giữa từ liên quan và không liên quan.
- **Ngưỡng downsample (`sample`=10^{-5}):** Giảm tần suất các từ phổ biến như "và", "của".
- **Số worker (`workers`=14):** Tận dụng đa luồng CPU để huấn luyện nhanh hơn.

Cơ chế **EpochSaver** (`CallbackAny2Vec`) được triển khai để tự động lưu checkpoint sau mỗi epoch.

3 Kết quả và Thảo luận

3.1 Kết quả định tính

Table 2: Kết quả tìm 5 từ gần nhất (Most Similar)

Từ gốc	**5 Từ gần nhất (Từ - Độ tương đồng)**	**Ghi chú**
`bitcoin`	`cash_bch` (0.7919), `đồng_litecoin` (0.7917), `tiền_điện_tử` (0.7770)	Mối quan hệ Crypto/Tài
`nhà_khoa_học`	`nhà_khoa_học_đa` (0.7026), `người_nhà_khoa_học` (0.6868), `avi_loeb` (0.6723)	Mối quan hệ nghề nghiệp
`hà_nội`	`trưng_thanh_xuân` (0.6557), `hà_nội_các` (0.6529), `trúc_bạch_tối` (0.6412)	Các địa danh lân cận
`vui_vẻ`	`vui_vè` (0.6180), `nạnh_tị` (0.6164), `hdv_phú_bình` (0.6119)	Kết quả kém chính xác h

1. Tìm từ gần nhất (Most Similar)

2. Phép toán vector (Word Analogy) Phép toán $A - B + C =?$ được thực hiện để đánh giá khả năng nắm bắt mối quan hệ logic.

- **Phép toán thành công:** `nhật_bản` - `hà_nội` + `việt_nam` = ?
- **Kết quả:** `nhật_bả` (0.5575), `cường_quốc_nhật_bản` (0.5045), ...
- **Thảo luận:** Mặc dù không tìm được thành phố Tokyo, kết quả vẫn xoay quanh chủ đề Nhật Bản.

3. Từ khác biệt (Odd One Out)

- `[hồ_chí_minh, đà_nẵng, phở, hải_phòng]` → **phở**
- `[bóng_đá, bơi, nhảy_múa, quần_vợt]` → **nhảy_múa**
- `[bàn_phím, màn_hình, cái_ghế, chuột]` → **chuột**

3.2 Đánh giá Tóm tắt Vector

- **Vector trung bình** của: `ô_tô`, `xe_máy`, `xe_đạp`.
- **Kết quả gần nhất:** `xe_máy` (0.8578), `ô_tô` (0.8056), `xe_đạp` (0.7723), `gáixe` (0.7715)

4 Kết luận và Công việc Tương lai

Nghiên cứu đã xây dựng thành công một mô hình nhúng từ Word2Vec tiếng Việt với vector 300 chiều trên một corpus dữ liệu lớn.

Công việc tương lai:

1. Tiếp tục train nốt các epoch còn lại.
2. Điều chỉnh `min_count` hoặc tiền xử lý từ đơn.
3. Thử nghiệm mô hình **FastText**.
4. Đánh giá trên các tác vụ xuôi dòng như Text Classification hoặc NER.

Tài liệu tham khảo

References

[1] Link code vector space model GitHub của em. `https://github.com/chiendz11/NLP/tree/master/week5`

[2] Vietnamese Natural Language Processing Toolkit. `https://github.com/undertheseanlp/underthesea`.

[3] Link chứa chi tiết thông corpus `https://huggingface.co/datasets/VTSNLP/vietnamese_curated_dataset`.