WEEK 3_ EMBEDDING EVALUATION

1. Tổng quan

a) Những điều đã hiểu và biết thêm được từ paper

Định nghĩa và vai trò

- Word embeddings: biểu diễn vector dense, low-dimensional cho từ.
- Giúp mô hình tận dụng tính chất các từ có ngữ cảnh giống nhau thì gần nhau trong không gian vector.

Hai phương pháp chính

- **Prediction-based models (Word2Vec)**: dự đoán từ dựa trên ngữ cảnh hoặc ngược lại.
  - ❖ Continuous Bag of Words (CBOW): dự đoán từ trung tâm dựa trên các từ ngữ cảnh xung quanh.
  - ❖ Skip-gram: dự đoán các từ ngữ cảnh dựa trên từ trung tâm dựa trên xác suất và tham số window size.
- **Count-based models (TF-IDF):** tính toán mức liên quan về mặt ngữ nghĩa giữa các từ bằng cách thống kê số lần đồng xuất hiện của một từ so với các từ khác.

Mối liên hệ giữa hai nhóm

- Một số nghiên cứu chỉ ra prediction-based (như Skip-gram Negative Sampling) có thể xem như dạng factorization của ma trận co-occurrence.
- Cho thấy ranh giới giữa hai hướng không hoàn toàn tách biệt.

Các hướng phát triển và mở rộng

- Word embeddings có thể được huấn luyện đặc thù cho từng tác vụ cụ thể (task-specific).
- Có thể mở rộng từ từ lên cụm từ, câu, tài liệu (compositional embeddings).
- Có tiềm năng kết hợp điểm mạnh của cả hai nhóm.

b) Những điều chưa hiểu

- Paper nói Negative Sampling giúp huấn luyện nhanh và hiệu quả, nhưng làm sao biết số lượng negative samples bao nhiêu là đủ để vừa tiết kiệm tính toán vừa giữ chất lượng embedding?
- Nếu ta thay đổi window size nhưng giữ nguyên số lượng negative samples, kết quả embedding có thay đổi đáng kể không và vì sao?
- Paper có nhắc đến Noise Contrastive Estimation (NCE) và Negative Sampling. Hai phương pháp này khác nhau ở điểm nào về ý tưởng và tác động đến chất lượng embedding?

2. **Embedding trong kiến trúc Transformer:**

Embeddings vẫn tồn tại trong Transformers. Mỗi từ đầu vào được chuyển thành token embedding + positional embedding để mô hình biết vị trí. Sau khi đi qua nhiều lớp self-attention, các embeddings này trở thành contextual embeddings. Cuối cùng, ta có thể lấy một embedding đại diện cho toàn bộ câu (ví dụ vector [CLS] trong BERT) để dùng cho tác vụ sau.

3. **Embedding Evaluation:**

3.1 Mục tiêu đánh giá:

- Embedding tốt cần mã hoá được quan hệ ngữ nghĩa/cú pháp để các điểm gần nhau trong không gian có ý nghĩa giống/ liên quan và thường đo bằng cosine similarity.
- Embedding cũng nên thể hiện được quan hệ tuyến tính cho phép loại suy (vector arithmetic) như "king − man + woman → queen," minh hoạ bằng việc truy vấn lân cận gần nhất theo cosine.

3.2 Đánh giá nội tại (intrinsic):

- Kiểm tra lân cận gần nhất: dùng cosine similarity và hàm most_similar để xem các từ láng giềng có hợp lý ngữ nghĩa không.(ví dụ man/woman, boy/girl) và tách khỏi từ khác loại như water.
- Phép tương tự (analogies): kiểm tra các phép cộng-trừ vector kinh điển như "king - man + woman ≈ queen" để thăm dò cấu trúc quan hệ ngữ nghĩa mà embeddings biểu hiện.
- Quan sát cấu trúc chiều: dùng đối chiếu màu/hoa văn để phát hiện cột/chiều chung giữa các từ cùng loại hoặc thuộc tính (ví dụ cột "đỏ" chạy xuyên qua nhiều từ người, khác với vật).

3.3 Đánh giá ngoại tại (extrinsic):

- Dùng embeddings làm đặc trưng cho tác vụ downstream (phân loại, gợi ý, mô hình ngôn ngữ bước 2) để xác nhận ích dụng thực tế vượt qua độ tương tự hình thức.
- Ghi nhận kinh nghiệm triển khai công nghiệp như hệ khuyến nghị/chuỗi thời gian tận dụng Word2Vec, như một chỉ báo hiệu quả trong môi trường sản xuất.

3.4 Đánh giá siêu tham số:

- Ưu tiên khảo sát hệ thống với hai nút chỉnh chính là cửa sổ ngữ cảnh và số negative samples, vì chúng chi phối trực tiếp "loại" tương tự và chất lượng lân cận/analogies quan sát được.
- VD: window nhỏ 2–15 thiên về hoán-đổi; window lớn 15–50+ thiên về liên-quan; negative 5–20 tốt, 2–5 đủ khi dữ liệu lớn; Gensim mặc định window=5, negative=5.

3.5 Độ phủ từ vựng và n-gram

- Kiểm tra min_count, max_final_vocab, sample và trim_rule để đảm bảo các thuật ngữ miền đều có vector; nếu thiếu phủ sẽ làm sai lệch đánh giá.
- Cân nhắc phát hiện phrase để học multiword expressions (Phrases) và so sánh chất lượng lân cận/analogies trước–sau khi đưa n-gram.

3.6 Tái lập và theo dõi

- Cố định seed, khi cần so sánh, đặt workers=1 và thiết lập PYTHONHASHSEED để giảm nhiễu lịch luồng OS, bảo đảm so sánh công bằng.
- Bật compute_loss và dùng get_latest_training_loss theo dõi hội tụ; với hs=1 và negative=0.

3.7 Hiệu năng và triển khai:

- Ưu tiên mô hình dễ stream và tối ưu C như Gensim để thử sai nhanh trên corpora lớn trong giai đoạn đánh giá.
- Lưu chỉ KeyedVectors và dùng mmap để giảm RAM, tăng tốc nạp/chia sẻ tiến trình; đồng thời hiểu giới hạn: C format không tiếp tục huấn luyện được.

4. **Bài tập: So sánh word embedding của 3 sentiments (positive, negative và neutral).**

- **Dataset:** Sp1786/multiclass-sentiment-analysis-dataset
- **Cách làm:** Code from scratch Word2Vec và sử dụng với từng từ riêng lẻ rồi tính trung bình cho cả câu với từng loại sentiment (positive, negative) và so sánh.
- **Đánh giá:**
  + Cosine Similarity Average
- **Chi tiết ở repo:** https://github.com/trantranuit/Embedding_Evaluation