BPE

1. Giới Thiệu Về BPE

Byte Pair Encoding (BPE) là một thuật toán mã hóa subword (từ con) được sử dụng rộng rãi trong xử lý ngôn ngữ tự nhiên (NLP), đặc biệt là trong các mô hình ngôn ngữ lớn như GPT. BPE giúp tách văn bản thành các đơn vị nhỏ hơn từ (subwords) bằng cách học từ dữ liệu huấn luyện, giảm kích thước từ vựng và xử lý tốt các từ hiếm hoặc từ mới.

- Mục đích chính: Giải quyết vấn đề từ vựng lớn trong tokenization bằng cách phân tích văn bản thành các cặp byte phổ biến nhất, thay vì sử dụng từ nguyên vẹn.
- Ưu điểm:
    - Xử lý từ out-of-vocabulary (OOV) bằng cách phân tích thành các subword quen thuộc.
    - Giảm số lượng token cần thiết, giúp mô hình học hiệu quả hơn.
    - Áp dụng cho nhiều ngôn ngữ, bao gồm tiếng Việt với các từ ghép phức tạp.
- Nhược điểm: Có thể tạo ra các token không có ý nghĩa ngôn ngữ rõ ràng, dẫn đến độ dài chuỗi token dài hơn so với tokenization dựa trên từ.

BPE được giới thiệu bởi Sennrich et al. (2016) trong lĩnh vực dịch máy, và sau đó được phổ biến bởi OpenAI trong các mô hình như GPT-2 và GPT-3.

2.Vấn Đề OOV Và Lợi Ích BPE

Tokenization theo từ dễ gặp OOV (từ mới không trong vocabulary). BPE phân tích từ thành subword quen thuộc, ví dụ: "lowest" → "low" + "est".

Ưu điểm: Xử lý từ hiếm, đa ngôn ngữ (như tiếng Việt với từ ghép); giảm token dài.

Nhược điểm: Subword có thể không ý nghĩa.

3.Các Thành Phần Tokenizer BPE

- **Pre-tokenizer:** Tách theo khoảng trắng, thêm "_" đánh dấu kết thúc từ.
- **Learner:** Học merges từ corpus.
- **Encoder/Decoder:** Chuyển văn bản thành ID token và ngược lại.

4.Thuật Toán BPE (Vòng Lặp Greedy)

1. Chuẩn Bị Corpus: Chuyển từ thành ký tự + "_". Ví dụ từ video (tần suất):
    - low (2): l o w _
    - lowest (1): l o w e s t _
    - new (1): n e w _
    - wider (3): w i d e r _
2. Vocabulary ban đầu: {_, e, d, i, l, n, o, r, s, t, w}.
3. Đếm Và Merge (lặp k lần, ví dụ 10.000-50.000):

- Đếm cặp phổ biến: (l, o): 3, (o, w): 3, (w, e): 4, (e, r): 4, v.v.
- Merge phổ biến nhất, ví dụ (l, o) → "lo"; cập nhật corpus: lo w _ (x2), lo w e s t _, ...
- Tiếp: Merge (lo, w) → "low"; (w, i) → "wi", v.v.
- Kết quả: Vocabulary mở rộng với subword như "low", "er", "est".

5.Công thức của thuật toán

- Khởi tạo từ điển V chứa tất cả các ký tự của ngữ liệu
- Lặp từ 1 đến k.
    - Tìm ra cặp 2 token xuất hiện nhiều nhất trong ngữ liệu
    - Nối 2 token này thành token mới token_new
    - Thêm token_new vào từ điển V
    - Thay thế cặp 2 token ban đầu bằng token_new ở mọi nơi trong ngữ liệu
- Kết quả thu được từ điển V

6.Encode Và Decode

- Encode: l o w e s t → l o w e s t _ → Áp dụng merges: low e s t _ → ["low", "est",...] → ID [12,...].
- Decode: Nối token, loại "_": "low" + "est" → "lowest".