announcement

Harito ID

2025-09-16

Thông báo nội dung buổi thực hành tuần này

Chào các bạn sinh viên,

Tuần này, chúng ta sẽ cùng nhau thực hành hai bài Lab quan trọng trong môn Xử lý Ngôn ngữ Tự nhiên (NLP): **Lab 1: Text Tokenization** và **Lab 2: Count Vectorization**. Đây là những kiến thức nền tảng để các bạn có thể xử lý và biểu diễn dữ liệu văn bản cho các tác vụ NLP sau này.

Nội dung thực hành

Lab 1: Text Tokenization (Tách từ/Tách token)

Trong Lab này, các bạn sẽ:

- Tìm hiểu về khái niệm và tầm quan trọng của việc tách từ (tokenization) trong NLP.
- Tự tay cài đặt một bộ tách từ đơn giản (`SimpleTokenizer`) dựa trên khoảng trắng và xử lý dấu câu cơ bản.
- (Bonus) Cài đặt một bộ tách từ nâng cao hơn (`RegexTokenizer`) sử dụng biểu thức chính quy để xử lý các trường hợp phức tạp hơn.
- Áp dụng các bộ tách từ đã cài đặt lên một phần của tập dữ liệu thực tế `UD_English-EWT` để quan sát và so sánh kết quả.

Lab 2: Count Vectorization (Biểu diễn vector đếm)

Tiếp nối Lab 1, trong Lab này, các bạn sẽ:

- Tìm hiểu cách biểu diễn văn bản dưới dạng vector số học bằng mô hình Bag-of-Words (Túi từ).
- Cài đặt lớp `CountVectorizer` để chuyển đổi một tập hợp các văn bản (corpus) thành ma trận tần số từ (document-term matrix). `CountVectorizer` này sẽ sử dụng bộ tách từ mà các bạn đã xây dựng ở Lab 1.
- Áp dụng `CountVectorizer` lên tập dữ liệu `UD_English-EWT` để thấy cách văn bản được chuyển đổi thành các vector số, sẵn sàng cho các mô hình học máy.

Chuẩn bị trước buổi học

Để buổi thực hành diễn ra hiệu quả, các bạn vui lòng:

1. Đảm bảo đã cài đặt đầy đủ các thư viện cần thiết trong `requirements.txt` bằng lệnh `pip install -r requirements.txt`.

2. Đọc qua nội dung của `lab/lab1_tokenization.md` và `lab/lab2_count_vectorization.md` để nắm được mục tiêu và các bước thực hiện.
3. Chuẩn bị môi trường lập trình (Python, IDE) sẵn sàng.
4. **Lưu ý về ngôn ngữ lập trình:** Các ví dụ code trong bài giảng được trình bày bằng Python. Tuy nhiên, các bạn hoàn toàn có thể sử dụng bất kỳ ngôn ngữ lập trình nào khác mà bạn thành thạo, miễn là vẫn duy trì được cấu trúc hướng đối tượng (OOP) và triển khai đúng các chức năng chính đã được yêu cầu trong bài giảng (ví dụ: các interface `Tokenizer`, `Vectorizer` và các phương thức `tokenize`, `fit`, `transform`).

Nếu có bất kỳ thắc mắc nào, đừng ngần ngại đặt câu hỏi trong buổi học hoặc trên diễn đàn của lớp.

Hướng dẫn nộp bài

Để nộp bài thực hành, các bạn vui lòng thực hiện các bước sau:

1. **Code trên GitHub:**
    - Tạo một repository mới trên GitHub (hoặc sử dụng repository đã có).
    - Đảm bảo toàn bộ mã nguồn của bạn (bao gồm các file đã chỉnh sửa trong `src/preprocessing`, `src/representations`, `src/core/interfaces.py` và các file `main.py` hoặc test mà bạn đã tạo/chỉnh sửa) được đẩy lên repository này.
    - Repository phải ở chế độ **Public** để giảng viên có thể truy cập và chấm bài.
2. **File mô tả công việc (README.md hoặc Report.md):**
    - Trong repository GitHub của bạn, tạo một file mô tả chi tiết về công việc bạn đã làm. File này có thể là `README.md` hoặc `Report.md`.
    - Nội dung file cần bao gồm:
        - **Mô tả công việc:** Bạn đã thực hiện những gì trong Lab 1 và Lab 2 (ví dụ: cài đặt `SimpleTokenizer`, `RegexTokenizer`, `CountVectorizer`, cách bạn xử lý các trường hợp đặc biệt, v.v.).
        - **Kết quả chạy code:** Trình bày các kết quả khi bạn chạy các đoạn code ví dụ (ví dụ: output của các tokenizer trên các câu mẫu, output của CountVectorizer trên corpus mẫu và trên dataset `UD_English-EWT`). Bạn có thể chụp ảnh màn hình hoặc copy-paste output vào file này.
        - **Giải thích kết quả:** Phân tích và giải thích các kết quả bạn thu được. Ví dụ: so sánh sự khác biệt giữa `SimpleTokenizer` và `RegexTokenizer`, nhận xét về vocabulary và document-term matrix của `CountVectorizer`, những khó khăn gặp phải và cách bạn giải quyết.
3. **Nộp link GitHub:**
    - Sau khi đã đẩy code và file mô tả lên GitHub, hãy nộp đường link (URL) của repository GitHub của bạn lên hệ thống Classroom theo đúng thời hạn.

Chúc các bạn có một buổi thực hành hiệu quả!

Trân trọng,