

BÁO CÁO MILESTONE 1: DATA ACQUISITION

Đồ án: E-Commerce Search Engine (SEG301)

1. Thông tin chung

Tên dự án: SEG301 E-Commerce Search Engine

Nhóm thực hiện: Nhóm 2

Thành viên:

Trịnh Khải Nguyên (QE190129) - Crawler Lead (Ebay, Chợ Tốt)

Lê Hoàng Hữu (QE190142) - Crawler (Tiki, Shopee)

Ngô Tuấn Hoàng (QE190076) - Crawler (Tiki, Shopee)

2. Mục tiêu Milestone 1

Mục tiêu chính của giai đoạn này là xây dựng hệ thống thu thập dữ liệu (Web Crawler) ổn định, hiệu quả để thu thập lượng lớn dữ liệu sản phẩm từ các sàn thương mại điện tử lớn (Shopee, Tiki, Chợ Tốt, eBay). Dữ liệu sau khi thu thập phải được làm sạch, chuẩn hóa và lưu trữ ở định dạng thống nhất để phục vụ cho các giai đoạn Indexing và Ranking sau này.

Mục tiêu cụ thể:

Xây dựng Crawler cho 4 sàn: Shopee, Tiki, Chợ Tốt, eBay.

Xử lý các cơ chế chặn bot (Rate limiting, CAPTCHA, IP blocking).

Chuẩn hóa dữ liệu về schema chung (ProductItem).

Lưu trữ dữ liệu dạng JSON Lines (JSONL) để tối ưu hóa việc đọc/ghi.

Mục tiêu dữ liệu: Tiếp cận con số 1.000.000 documents.

3. Kết quả đạt được

3.1. Thống kê dữ liệu (Thực tế)

Đến thời điểm hiện tại, nhóm đã xây dựng thành công bộ Crawler và thu thập được khối lượng dữ liệu khổng lồ:

Sàn TMĐT	Số lượng (Docs)	Tỷ lệ đóng góp	Trạng thái
Shopee	800,284	55.0%	Hoàn thành
Tiki	435,203	29.9%	Hoàn thành
Chợ Tốt	114,370	7.9%	Hoàn thành
eBay	104,742	7.2%	Hoàn thành
Other	104,742	7.2%	Hoàn thành
TỔNG CỘNG	1,454,599	100%	Vượt chỉ tiêu 45%

3.2. Chất lượng dữ liệu

Unified Schema: Tất cả dữ liệu từ 4 sàn đều được map về cấu trúc JSON chung bao gồm các trường: `id`, `platform`, `title`, `price`, `original_price`, `url`, `image_url`, `rating_average`, `review_count`, `sold_count`, `brand`, `category`.

Data Cleaning:

Xử lý giá trị tiền tệ về số nguyên (Integer).

Loại bỏ HTML tags trong mô tả sản phẩm.

Chuẩn hóa văn bản (lowercase, xóa ký tự đặc biệt) phục vụ indexing.

Lưu trữ: Dữ liệu được lưu dạng `.jsonl` (mỗi dòng là 1 object JSON), giúp dễ dàng append dữ liệu mới và đọc từng dòng (memory efficient).

3.3. Insight Dữ liệu (Text Statistics)

Phân tích sơ bộ trên tập dữ liệu 1.45 triệu văn bản cho thấy độ phong phú của tập từ vựng:

Tổng số văn bản (Docs): 1,454,599 dòng

Tổng số từ (Total Words): 20,434,969 từ

Từ vựng (Vocab Size): 381,724 từ duy nhất

Độ dài trung bình (Avg Length): 14.05 từ/dòng

Nhận xét: Độ dài trung bình 14 từ/dòng là lý tưởng cho bài toán Search Engine e-commerce, vì tên sản phẩm thường ngắn gọn, cô đọng keyword quan trọng.

4. Giải pháp kỹ thuật

4.1 Kiến trúc Crawler

Ngôn ngữ: Python 3.10+

Thư viện chính:

`aiohttp / asyncio`: Sử dụng cho Chợ Tốt và các tác vụ yêu cầu tốc độ cao

(Asynchronous request).

requests : Sử dụng cho các crawler cơ bản (Synchnorous).

DriissionPage / Selenium : Xử lý các trang web dynamic (Shopee) có cơ chế chặn bot phức tạp.

Cơ chế hoạt động:

Input: Danh sách Category ID hoặc Keywords.

Processing: Gửi request (giả lập Headers, User-Agent) -> Nhận JSON/HTML -> Parse & Map dữ liệu -> Clean Data.

Output: Ghi nối đuôi (Append) vào file `.jsonl`.

4.2 Xử lý dữ liệu (Data Processing)

Sau khi thu thập, dữ liệu thô (Raw Data) được đưa qua pipeline xử lý tập trung để đảm bảo chất lượng:

Chuẩn hóa (Normalization):

Script: `normalize_data.py`

Mapping: Đồng bộ các tên trường khác nhau (ví dụ: `name`, `subject`) về trường chuẩn `title`.

Price Cleaning: Sử dụng Regex để tách số từ chuỗi giá tiền (VD: "1.200.000đ" -> `1200000`), loại bỏ dấu chấm/phẩy.

ID Uniformity: Đảm bảo tất cả ID đều có tiền tố sàn (VD: `shopee_12345`) để tránh trùng lặp giữa các sàn.

Làm sạch (Cleaning):

Validation: Loại bỏ các bản ghi rác bị thiếu `id` hoặc `title`.

Text Cleaning: Loại bỏ các thẻ HTML dư thừa, chuyển đổi `categories` về dạng danh sách (List).

Khử trùng lặp (De-duplication):

Script: `deduplicate.py`

Logic: Kiểm tra trùng lặp dựa trên `id`. Nếu tìm thấy ID trùng, hệ thống sẽ giữ lại bản ghi mới nhất dựa trên timestamp `crawled_at`.

Kết quả: Loại bỏ hàng nghìn bản ghi trùng lặp do quá trình crawl chồng chéo hoặc crawl lại nhiều lần.

4.3 Chiến lược Anti-Bot (Chống chặn)

Fake User-Agent: Xoay vòng User-Agent để giả lập các trình duyệt khác nhau.

Delay Request: Thêm thời gian nghỉ ngẫu nhiên (`random.sleep`) giữa các request để tránh bị flag là bot.

API Reverse Engineering: Ưu tiên tấn công vào API ẩn (Mobile API/Internal API) của sàn thay vì parse HTML giao diện, giúp tốc độ nhanh hơn và dữ liệu sạch hơn.

5. Khó khăn & Giải pháp

Khó khăn	Giải pháp đã thực hiện
Shopee chặn IP/Request gắt gao	Chuyển sang sử dụng DrissionPage để điều khiển trình duyệt thật, kết hợp xoay vòng Proxy nếu cần thiết.
Dữ liệu phân mảnh, khác cấu trúc	Xây dựng file schema.py dùng chung, buộc tất cả crawler phải convert dữ liệu về chuẩn ProductItem trước khi lưu.
Tốc độ crawl chậm khi scale lớn	Áp dụng kỹ thuật bất đồng bộ (asyncio) cho phép gửi hàng trăm request cùng lúc (đối với Chợ Tốt/Tiki).

6. Kế hoạch tiếp theo (Milestone 2)

Trong giai đoạn tới (Core Search Engine), nhóm sẽ tập trung vào:

Indexing: Triển khai thuật toán **SPIMI (Single-Pass In-Memory Indexing)** để tạo chỉ mục ngược (Inverted Index) từ bộ dữ liệu đã thu thập.

Ranking: Tự code thuật toán **BM25** để xếp hạng kết quả tìm kiếm dựa trên độ phù hợp.

Optimization: Tối ưu hóa bộ nhớ và tốc độ truy vấn cho index của 1 triệu documents.

Unit Test: Viết test case để kiểm chứng độ chính xác của thuật toán Ranking (so với thư viện chuẩn).

7. Kết luận (Conclusion)

Milestone 1 đã hoàn thành với việc vượt chỉ tiêu thu thập dữ liệu, đảm bảo chất lượng và tính nhất quán cao. Hệ thống Crawler hoạt động ổn định, xử lý tốt các cơ chế chặn bot phức tạp từ Shopee và Tiki. Với kho dữ liệu sạch và cấu trúc chuẩn hóa này, nhóm đã sẵn sàng toàn diện để bước vào giai đoạn then chốt tiếp theo: Xây dựng Core Search Engine (Indexing & Ranking).