

BÁO CÁO MILESTONE 1: DATA ACQUISITION

Đồ án: E-Commerce Search Engine (SEG301)

1. Thông tin chung

Thông tin	Chi tiết
Tên dự án	SEG301 E-Commerce Search Engine
Nhóm thực hiện	Nhóm 2
Giai đoạn	Milestone 1 - Data Acquisition

Thành viên nhóm

STT	Họ và tên	MSSV	Vai trò
1	Trịnh Khải Nguyên	QE190129	Crawler Lead (eBay, Chợ Tốt)
2	Lê Hoàng Hữu	QE190142	Crawler (Tiki, Shopee)
3	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. 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
TỔNG CỘNG	1,454,599	100%	Vượt 45%

3.2. Chất lượng dữ liệu

Unified Schema: Tất cả dữ liệu từ 4 sàn đều được map về cấu trúc JSON chung:

Trường	Kiểu dữ liệu	Mô tả
<code>id</code>	String	ID duy nhất, format: <code>{platform}_{original_id}</code>
<code>platform</code>	String	Tên sàn TMĐT (lowercase)
<code>title</code>	String	Tên sản phẩm
<code>price</code>	Integer	Giá bán hiện tại
<code>original_price</code>	Integer	Giá gốc (trước giảm giá)
<code>sold_count</code>	Integer	Số lượng đã bán
<code>link</code>	String	URL sản phẩm
<code>image_url</code>	String	URL hình ảnh
<code>category</code>	String	Danh mục sản phẩm
<code>brand</code>	String	Thương hiệu

Data Cleaning đã thực hiện:

- 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:

Chỉ số	Giá trị
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:

Thư viện	Mục đích sử dụng
aiohttp / asyncio	Sử dụng cho Chợ Tốt và các tác vụ tốc độ cao (Asynchronous)
requests	Sử dụng cho các crawler cơ bản (Synchronous)
DriissionPage / Selenium	Xử lý trang web dynamic (Shopee) có cơ chế chặn bot phức tạp
BeautifulSoup	Parse HTML (eBay)

Cơ chế hoạt động:

- Input:** Danh sách Category ID hoặc Keywords
- Processing:** Gửi request → 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:

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)

- **ID Uniformity:** Đảm bảo tất cả ID đều có tiền tố sàn (VD: shopee_12345)

Làm sạch (Cleaning) - Script: `cleaner.py`

- **Validation:** Loại bỏ các bản ghi rác bị thiếu id hoặc title
- **Text Cleaning:** Loại bỏ các thẻ HTML dư thừa

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
 - **Kết quả:** Loại bỏ hàng nghìn bản ghi trùng lặp do quá trình crawl chòng chéo
-

4.3. Chiến lược Anti-Bot (Chống chặn)

Kỹ thuật	Mô tả
Fake User-Agent	Xoay vòng User-Agent để giả lập các trình duyệt khác nhau
Delay Request	Thêm thời gian nghỉ ngẫu nhiên (<code>random.sleep</code>) giữa các request
API Reverse Engineering	Ưu tiên tấn công vào API ẩn (Mobile API/Internal API) thay vì parse HTML giao diện

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 <code>schema.py</code> dùng chung, buộc tất cả crawler phải convert dữ liệu về chuẩn <code>ProductItem</code> trước khi lưu
Tốc độ crawl chậm khi scale lớn	Áp dụng kỹ thuật bắt đồng bộ (<code>asyncio</code>) 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:

STT	Công việc	Mô tả
1	Indexing	Triển khai thuật toán SPIMI (Single-Pass In-Memory Indexing) để tạo chỉ mục ngược (Inverted Index)
2	Ranking	Tự code thuật toán BM25 để xếp hạng kết quả tìm kiếm dựa trên độ phù hợp
3	Optimization	Tối ưu hóa bộ nhớ và tốc độ truy vấn cho index của 1 triệu documents
4	Unit Test	Viết test case để kiểm chứng độ chính xác của thuật toán Ranking

7. Kết luận

Milestone 1 đã hoàn thành với việc **vượt chỉ tiêu thu thập dữ liệu 45%**, 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)**.

Báo cáo được tạo bởi Nhóm 2 - SEG301