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	Leader (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, 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

A. Unified Schema (Cấu trúc chuẩn hóa)

Dữ liệu từ 4 sàn được ánh xạ về cấu trúc chung `ProductItem` (định nghĩa trong `schema_shared.py`).

Bảng 1: Thông tin Sản phẩm (Business Info)

Trường	Kiểu	Mô tả
`id`	String	Format: `{platform}_{hash}` (Duy nhất)
`platform`	String	Tên sàn: `shopee`, `tiki`, `ebay`…
`title`	String	Tên gốc của sản phẩm
`price`	Int	Giá bán (Đã ép kiểu số nguyên)
`original_price`	Int	Giá gốc trước giảm
`sold_count`	Int	Số lượng bán (VD: `5200`)
`category`	String	Danh mục sản phẩm
`brand`	String	Thương hiệu

Bảng 2: Thông tin Kỹ thuật (System Info)

Trường	Kiểu	Mô tả
`link`	String	URL sản phẩm sạch (bỏ tracking)
`image_url`	String	Link ảnh đại diện
`title_clean`	String	Tên đã làm sạch (lowercase, no icon)
`title_segmented`	String	Tên đã tách từ (dùng cho Indexing)
`source_file`	String	Nguồn file dữ liệu gốc

B. Các bước làm sạch dữ liệu (Data Cleaning):

1. **Xử lý tiền tệ:** Chuyển đổi giá trị về số nguyên (Integer).
2. **Làm sạch văn bản:** Loại bỏ HTML tags, icon rác trong mô tả/tiêu đề.
3. **NLP:** Tích hợp thư viện **PyVi** để tách từ tiếng Việt.
4. **Lưu trữ:** Dữ liệu được lưu dạng `.jsonl` .

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)
`DrissionPage` / `Selenium`	Xử lý trang web dynamic (Shopee) có cơ chế chặn bot phức tạp
`BeautifulSoup`	Parse HTML (eBay)
`Multiprocessing`	Hỗ trợ chạy đa tiến trình (Tiki)
`PyVi`	Sử dụng để tách từ

4. Cơ Chế Hoạt Động (Operational Mechanism)

1. Thu thập dữ liệu (Data Acquisition)

Quy trình thu thập vận hành theo cơ chế **Hybrid (Lai ghép)** để tối ưu hóa cho từng nền tảng:

1. **Input:**
   - Danh sách **Category ID** (đối với Tiki).
   - Danh sách **Keywords** từ khóa (đối với Shopee).
2. **Processing (Xử lý song song):**

- **Luồng Tiki:** Sử dụng kỹ thuật *API Reverse Engineering* kết hợp *Multi-threading* để gửi request tốc độ cao, trích xuất dữ liệu JSON trực tiếp.
- **Luồng Shopee:** Sử dụng kỹ thuật *Browser Automation* (với thư viện DrissionPage) kết hợp *Multi-processing* để điều khiển trình duyệt thật. Hệ thống tự động cuộn trang (Lazy load handling), vượt Captcha và bóc tách dữ liệu từ HTML.

3. **Output:**
   - Dữ liệu được ghi thời gian thực (Real-time logging) vào các file phân tán `.jsonl` trong thư mục `raw_data/`.

2. Xử lý dữ liệu (Data Processing)

Sau khi thu thập, toàn bộ dữ liệu thô (Raw Data) được đưa qua pipeline xử lý tập trung để tạo ra bộ dữ liệu Master:

A. Chuẩn hóa (Normalization)

- **Script:** `clean_merged_data.py`
- **Schema Mapping:** Đồng bộ các tên trường khác nhau (ví dụ: `url` của Tiki, `product_url` của eBay) về trường chuẩn `link` theo cấu trúc `ProductItem`.
- **Platform Detection:** Tự động nhận diện và gán nhãn sàn TMĐT (Shopee, Tiki, Lazada, eBay, Chợ Tốt) dựa trên phân tích cấu trúc URL.
- **Type Casting:** Ép kiểu dữ liệu an toàn để phục vụ tính toán (VD: Chuyển đổi `sold` "20k+" → `20000` integer).
- **ID Generation:** Tạo ID định danh duy nhất (Hashing) cho các sản phẩm thiếu ID gốc.

B. Làm sạch & NLP (Cleaning & Segmentation)

- **Script:** `clean_merged_data.py`
- **Text Cleaning:** Loại bỏ các ký tự đặc biệt, thẻ HTML rác, icon không cần thiết trong tiêu đề và mô tả.
- **Word Segmentation (Tách từ):** Tích hợp thư viện **PyVi** để xử lý ngôn ngữ tự nhiên tiếng Việt (VD: "máy giặt" → `máy_giặt`), giúp tăng độ chính xác cho giai đoạn Indexing sau này.
- **Validation:** Lọc bỏ các bản ghi rác, thiếu thông tin quan trọng (Null/Empty rows). #### **C. Khử trùng lặp (De-duplication)**
- **Script:** `clean_merged_data.py`
- **Logic:** Sử dụng cấu trúc dữ liệu `Set` để kiểm tra trùng lặp dựa trên khóa chính (`product_id` hoặc `link_hash`).
- **Cơ chế:** Khi gộp (Merge) dữ liệu từ nhiều nguồn/nhiều máy cào, hệ thống ưu tiên giữ lại bản ghi đầu tiên và loại bỏ các bản sao xuất hiện sau.
- **Kết quả:** Tạo ra file `MASTER_DATA_CLEAN.jsonl` duy nhất, sạch và đồng nhất cấu trúc.

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 chồng 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 (`random.sleep`) 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 `schema.py` dùng chung, buộc tất cả crawler phải convert dữ liệu về chuẩn `ProductItem` trước khi lưu

Khó khăn	Giải pháp đã thực hiện
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:

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