

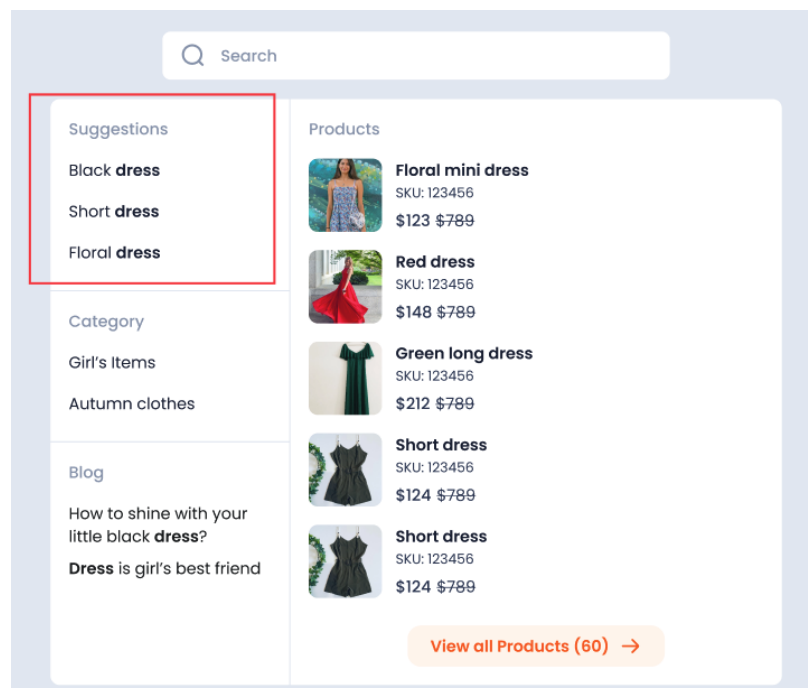
[Spec] Autocomplete Suggestions

Tổng quan:

Khi khách hàng enable tính năng Instant Search Widget, mỗi khi KH typing ký tự thì app sẽ chủ động suggest autocomplete cho KH các keyword.

VD:

KH type từ dress -> hệ thống sẽ suggest: black **dress**, short **dress**, floral **dress**,...



Source index:

1. Product data:

- Dữ liệu được collect trực tiếp từ product data trong store với logic như sau:
 - Các product/category/blog field với hàm lượng từ ngữ lớn sẽ được tiến hành đưa vào hệ thống phân loại (product's name, description,...)
 - Hệ thống sẽ filter các câu dài thành các keyword ngắn gọn (VD: there is a massive volume of knowledge inside the book -> 'massive volume', 'knowledge', 'book')
- Dữ liệu này sẽ được tự động xoá và nạp mới sau 3 tháng

2. Search query data:

- Dữ liệu được tổng hợp từ 5000 search query gần nhất của KH. Rule lấy search query:

- Lưu search query khi người dùng thực hiện truy cập Search Result Page bằng query đó
- Không tạo ra "Did You Mean?" trong vòng 24H trước đó (ko có kết quả search)
- Chỉ được phép chứa các standard alphanumeric characters, spaces, slashes, single/double quotes, hyphens, periods, number signs, and percent signs.
- Chứa từ 3 đến 70 ký tự (bao gồm dấu cách)
- Không được phép dài quá 10 từ
- Không có từ nào dài hơn 20 ký tự. Trừ khi bản thân trong từ đó chứa dấu "," hoặc "-" (VD: 3,5-Dinitrobenzoyl")
- Dữ liệu này sẽ được tự động xóa và nạp mới sau 1 tháng

Logic xử lý data

1. Product data

- Các information đến từ product sẽ được xử lý riêng biệt theo logic sau:
 - Product name + Product variant: Thực hiện tách từ name và variants thành các keyword, sau đấy các keyword được gộp lại với nhau thành 1 keyword lớn hơn, đánh số theo logic gán điểm (mô tả ở dưới) và lưu vào database
 - Ví dụ: **[Sample] Fog Linen Chambray Towel - The Beige Stripe**
 - Output: **sample fog linen chambray towel beige stripe**
 - Product description: tách từ description thành các keyword riêng biệt, các keyword này sẽ được đánh số theo logic gán điểm (mô tả ở dưới) và lưu vào database.
 - Ví dụ: **The last laundry cart you'll ever buy. This industrial non-collapsible cart is constructed with a steel framed body using heavyweight canvas capped with a soft leather rim sitting on four smooth rolling casters.**
 - **last laundry cart buy**
 - **industrial non-collapsible cart constructed steel framed body heavyweight canvas capped soft leather rim sitting four smooth rolling casters**

2. Search query data

- Các query sẽ được đánh số theo logic gán điểm và lưu vào database

Logic hiển thị suggestions:

Bắt buộc phải là keyword đứng đầu trong 1 từ. VD: KH search "**dr**" -> app suggest "**d**ress", không phải "ear**d**rops".

Thứ tự suggest sẽ được hiển thị theo 3 mức UT dưới đây:

1. Exact match:

- Dữ liệu từ Search query hoặc Product data mà đúng 100% với search keyword của KH (VD: Exact SKU match)

2. Search query:

- Hệ thống "auto suggest" để hoàn thiện tiếp cụm text còn thiếu trong câu query của KH (so sánh với dữ liệu 5000 search query đã import)
- Thứ tự hiển thị như sau:
 - Ưu tiên query điểm cao hơn hiện ra trước (logic gán điểm được mô tả ở dưới)
 - Ưu tiên query ngắn hơn hiện ra trước
- Khi hiển thị hết các suggestion

3. Product data:

- Hệ thống "auto suggest" để hoàn thiện tiếp cụm text còn thiếu trong câu query của KH (so sánh với dữ liệu product data đã import)
- Thứ tự hiển thị như sau:
 - Ưu tiên query điểm cao hơn hiện ra trước (logic gán điểm được mô tả ở dưới)
 - Ưu tiên query ngắn hơn hiện ra trước

Logic gán điểm:

1. Đối với Search query:

- Mỗi output search query suggest ra đều được gán điểm mặc định = 1
- Mỗi output suggest được KH click vào sẽ được +1 điểm

2. Đối với Product data:

- Mỗi output được autocomplete ra product name, category name -> gán điểm = 3
- Mỗi output được autocomplete ra variant, SKU -> gán điểm = 2
- Mỗi output được autocomplete ra description -> gán điểm = 1

Inferred Queries (Complete the sentence)

We look at this group of suggestions and see if the user's query has a prefix match for any words in the current suggestions, and create a new suggestion. This is intended to find a closer representation of a suggestion to what the user is typing - Attempting to complete the currently typed word.

Example - If the user types "**blue dress p**" and one of the existing suggestions is "**blue dress with poka dots**", we'll intelligently grab the next word from that phrase that starts with "p" and append it, resulting in "**blue dress poka**"