

LOOP54 PRODUCT SEARCH AND CATEGORY NAVIGATION

Software Specification Sheet

Last updated: 2017-10-19

Table of contents

Product Feed	<u>Page 3</u>
Search	<u>Page 3</u>
Autocomplete	<u>Page 6</u>
Filters	<u>Page 7</u>
Facets	<u>Page 7</u>
Sorting	<u>Page 8</u>
Events	<u>Page 8</u>
Personalisation	<u>Page 8</u>
Other Logic	<u>Page 9</u>
Category Listings	<u>Page 11</u>
Versioning	<u>Page 11</u>
Performance and privacy	<u>Page 12</u>
Search Analytics	<u>Page 12</u>

Product Feed	
Automatic sync	Product feed is synced at least once a day. More frequently if needed.
Real-time delta sync and catalogue indexation	Delta-information/changes pushed to our API continuously in real-time, but it should always be complemented with a complete feed that Loop54 can synchronise with at least once a day. Loop54 can fetch changes or retailer can push changes.
Merge and sync different data sources and formats	If different information needs to be synced, it does not have to be in the same feed as products. For example, Loop54 can enrich the feed with data from other sources, like images, CRM or sales data. Data can also be in different formats. Engine can index different formats from different sources and merge them. (e.g. CSV, XML, JSON).
Search	
Related Results	In addition to Direct Results, the engine generates a list of Related Results. These Related Results are completely independent of the actual search query (i.e. the Related Results do not contain the search term(s) in their metadata), but are dependent rather on which products are considered Direct Results.
Related queries	Engine will display query suggestions that will lead to similar results as original query.

Search**Rank results by relevance**

By default, all search results - Direct and Related - returned by the engine are ranked by relevance. Relevancy is determined through Machine Learning, which combines keyword-matching, behavioural insights, business logic and the network of relationships between products. Learn more: loop54.com/how-product-search-works

Suggested spelling corrections

When the query contains spelling errors that the engine can interpret with some confidence, it will return spelling suggestions. The engine normally auto-corrects minor misspellings, but if the query is severely misspelled, then the engine will return "best guess" results and provide suggested spelling corrections. The engine is able to determine the most relevant spelling corrections by looking at past queries by other users that led to a direct hit (direct hit = search phrase exists, exactly as typed, in a product's metadata)

Search

**Fuzzy matching
(spellcheck)**

Minor misspellings are deciphered automatically and generate results without any human intervention. Results generated are the same as if the query was not misspelled. Loop54 will also find partial words (e.g. if query is for "bread" we also find "gingerbread").

**Fuzzy matching with
"MakeSense = False"**

If needed, the engine will edit the search query significantly. The engine tries to break a word down into several words, traversing back through the query until it finds a word that has a match in the catalogue. If the engine still can not find a match from sub-words, then it will drastically edit the search phrase (e.g. by swapping letters, replacing letters, etc.).

**Learning new words
(synonyms)**

Part of fuzzy matching. If fuzzy match is "Make Sense = False" and there is a frequent behaviour pattern which accompanies a "Make Sense = False" query, then the engine can learn the word. To use the feature "Make Sense = False", results must be displayed and behaviour events must be sent to Loop54.

SKU/EAN search

Only show one product as result if search is for SKU or EAN number.

**Content search
(extra index)**

Search for other things than products (e.g. ads, blogs, articles, etc.). List of content results can be returned in one list along with products or in separate list. If returned in one list, then both products and content are scored for relevancy against each other.

Autocomplete**Popular queries**

Autocomplete suggestions ranked by popularity. Popularity of suggestions are determined by the frequency of searches for that term. The term's popularity is built up by previous searches for that term that led to direct hits (direct hit = search phrase exists, exactly as typed, in a product's metadata)

Scoped autocomplete

Can use any facet to present autocomplete suggestions. Typically the facets used are brand or category. (e.g. "Prada in Accessories" or "Prada in Shoes")

Pre-populated autocomplete

Keyword extraction from brands and categories. At initial implementation there are no popular previous queries to list in the autocomplete. The engine can then give autocomplete suggestions by looking at any of the brands or categories were a match has been found. This makes sure that there is always autocomplete suggestions, starting from day one. Basically, the source for a cold-start autocomplete becomes brands and categories.

Keyword redirects
(COMING SOON)

If user enters a pre-defined keyword (e.g. opening hours), they are redirected to a specified URL immediately after pressing enter. The engine returns a link to use for the redirect.

Filters

Bespoke static sync filters

Engine can handle any static filters based on any attribute. These are filters that do not change between syncs of product feed. (e.g. out of stock, available in store, not yet released)

Bespoke dynamic request filters

Filters that are created based on live dynamic user information. Product assortment is filtered based on, for example, specific customer groups, geographic markets or specific physical stores. Can also be used for internal diagnostics.

Facets

Faceting on any product attribute

Faceted filters can be built using any product attribute (date, number or text). Manufacturer, category and price are standard.

Semi-dynamic faceted filters

Add and subtract facets simultaneously. As faceted filters are applied, the engine will remove the facets that no longer have a match, except for within the scope in which the user made their initial selection (e.g. if "Blue" is selected within the scope "Colours" then only items described as Blue will be displayed. However, the other colour filter options will remain visible with their previous count of matched items). By retaining the filter and the previous count of products within that filter, users can work their way back easily. **NOTE:** No new faceted filters will appear as search results are narrowed - this would make faceted filters fully-dynamic.

Sorting	
Sorting by popularity	Popularity can be determined by either Click, Add-To-Cart or Purchase (i.e. not a combination of the three). The engine can also use other data sources to determine popularity, such as Sales or CRM data. The engine can start with Sales data to overcome a "cold-start" and switch to a behaviour-based popularity ranking when enough behaviour data has been accumulated.
Sorting by any attribute	Sorting products by any attribute (e.g. popularity, price, newest/oldest, etc.).
Events	
Global behaviour events	Global events are the user behaviour events (e.g. clicks, add-to-cart, purchases, etc.) that occur across the entire website. These global events are tracked and used to improve the sorting of search results and category listings (i.e. to determine personal taste and popularity). The engine keeps track of where the event occurred - either within search or navigation.
Personalisation	
Sorting products on unique user identifier	Engine can sort results and category listings based on a user's unique personal taste, using browser cookie or user ID.
Cross device personalisation	Personalised experience can be reflected across all the devices a user might use. Only if implemented via login ID or user name, not with cookie-based tracking.

Other Logic

Boost and bury rules

Boost or bury results based on any product attribute. For example, promotional campaign, new products, high-margin products, etc. Boost and bury rules are applied to a relevant list of results/product.

Additional languages

Unlike many language-specific stemming algorithms that have a set of predefined rules of how to find the root form of a word, Loop54 handles things like stemming and other NLP tasks without any assumptions about how the language is built. Therefore multiple languages can be supported, but each language requires its own engine.

**Product information page
(product data page)**

Loop54 can be the source of product page information (e.g. variant images, product specifications, etc.). It is often the case that information for the product page is sourced faster from Loop54 than from the eCommerce platform.

Related products

This list appears on a product page and is made up of the products that are similar to the product being viewed. This list is created exactly the same way the engine locates Related results. This list is generated with a separate API call.

Segment specific results

Loop54 can limit the assortment of products shown in results or category listings based on specific customer groups, geographic markets, specific physical stores, or for the purpose of internal diagnostics. The criteria used to customise the assortment does change between syncs. Loop54 can source the groups and their respective assortment information in real-time by communicating directly with retailer's PIM and/or ecommerce platform.

Segment specific prices

Prices can adapt to the conditions of pre-set customer groups. Prices and customer groups need to be in the feed. Although the number of customer groups used is technically limitless, Loop54 imposes a soft upper limit of 100 customer groups. Loop54 can source the groups and their respective price information in real-time by communicating directly with retailer's PIM and/or ecommerce platform.

Other Logic

Rolling event window

The timeframe for event data collection can be customised (e.g. last 30 days, last 90 days, etc.) Shorter time frames often useful for more seasonal catalogues and products.

Parent-child variants

Parent/children connection ID's (used to associate articles and products and to make all metadata searchable even if only one variant is displayed in the search results). Option #1 - show parent product only. Option #2 - show all variants, not the parent. Option #3 - show either parent or variant depending on search breadth (i.e. if there are many hits, like more than one page of results, the engine will show the parent. And if there are few results, the engine will show all variants). Feed needs to include robust data (including product page url, product images, price, etc.) on all variants for options #2 and #3 to function.

Warm-up the search engine

Pre-populate algorithm logic with sales data and/or search phrases. At launch, the engine will know which products are the top sellers and will use this information to improve sorting of results. Or the engine will know which search phrases are most common and will use this information to pre-populate the autocomplete suggestions, spelling suggestions and related queries. This is done to overcome a cold-start problem. Typically this is only valuable for a few weeks until the engine has acquired enough of its own data.

Diffuse sorting of results

Option to not show items from the same shop/brand/campaign next to each other in search results.

Custom API

Any custom feature or function that:

- Changes the behaviour of a standard feature; AND
- Requires on-going maintenance, or has possibility to break in subsequent updates to algorithm or feature; OR
- Is a completely new feature - done in config or backend (i.e. a new API endpoint).

Used most often for complex data feeds or unique merchandising requirements.

Category Listings

Products sorted by general popularity

Engine will continuously learn what products (and network/context of products) are most popular. Popularity is based on global events like click, add-to-cart and purchase, and/or CRM information. Sorting is based on general popularity (i.e. the aggregate).

Products sorted by personal taste

Engine will continuously learn what products (and network/context of products) are most popular for a given user. Personal taste is based on the user's global events like click, add-to-cart and purchase, and/or CRM information for that unique user. Sorting is based on personal taste (i.e. the individual).

Use any business logic

Use boost and bury rules, parent/child variants, segment specific pricing and assortments, and dynamic filters in category listings as it is used in search.

Warm-up sorting

Using CRM, Sales data to sort category listings initially. Once enough behaviour data is accumulated the engine takes over.

Sort by any bespoke criteria

Engine can sort category listings by any bespoke criteria or attribute, like title, brand, price, etc.

Versioning

Automatic rollout of new engines

Loop54 is SaaS (software-as-a-service), this means our engines are hosted in the cloud and new versions can be rolled out without any extra work or cost. To access some new features, retailers may need to implement new APIs.

Performance/Privacy	
Monitoring	Monthly Uptime Percentage (MUP) of 99.98%. All network traffic is saved and analysed for volume and customer behaviour in MSSQL DB. Traffic data and server monitoring is sent to ELK (Elastic - Logstash - Kibana) system and Grafana where different stats like response time, sync, memory usage are monitored.
Cached responses	There is the possibility to cache responses.
Load distribution	Possibility to distribute load across multiple servers and work with load balancers.
Failover measures	Loop54 has failover proxies hosted in Sweden and Amsterdam
Data privacy	Encrypted SSL (HTTPS) and, if requested, IP-blocking to backend servers.
Search analytics	
Insights Dashboard	Unlike Google Analytics, which only provides unique searches, Loop54 will provide total search volume with adjustable time periods. The Insights Dashboard will also display analytics regarding search trends over time, changes in product rank order, and conversions rates for all queries (click, add-to-cart, purchase).
Analytics on "MakeSense False" results	Ability to see queries that the engine could not make sense of "as is" (i.e. the words do not exist anywhere in the catalogue). When "MakeSense" is False, the engine will edit the query to find a match. Being able to view search analytics on "MakeSense False" queries gives an indication of what new words the engine may have already learned or may eventually learn (i.e. in order for engine to learn these new words, "Make Sense False" results must be displayed to the user and behaviour events must be sent to Loop54).
Export data in XML	Loop54 provides API so all search insights collected by Loop54 can be exported. Only from the backend.



Contact Loop54

The Loop54 Group AB
Barnhusgatan 20
111 23 Stockholm, Sweden
+44 20 8133 9848, +46 70 185 9082
info@loop54.com
loop54.com