Deconstructing E-Commerce Search:

The 12 Query Types

You can always find the most up-to-date version of this article online at:
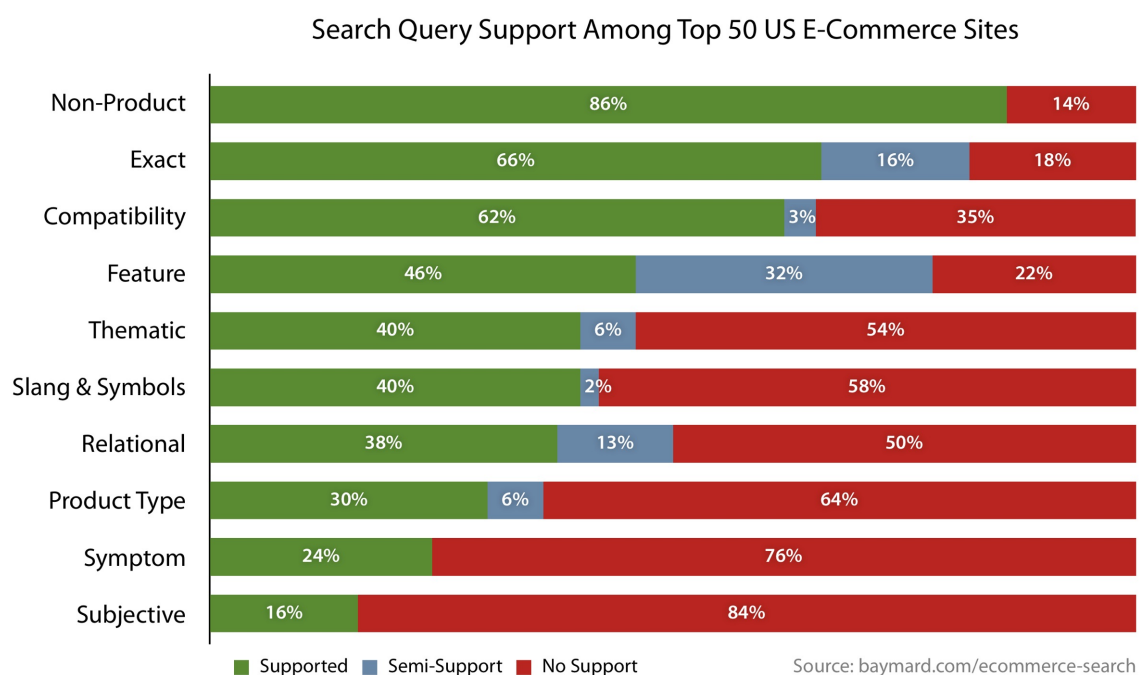
Published June 18, 2014 by Baymard Institute

The 12 Query Types

In this article we'll introduce 12 types of search queries identified during our large-scale usability study of e-commerce search. While not exhaustive they reflect the main types of queries that users rely on when searching in an e-commerce context.

During the usability study, the test subjects were observed to rely heavily on e-commerce search queries that included a theme, feature, relation, or symptom – yet most of 19 tested sites had poor support for these query types. Even among the 50 e-commerce giants we've benchmarked the support for most query types is meager at best, as should be evident from the graph below.

Search Query Support Among Top 50 US E-Commerce Sites

Query Type	Supported	Semi-Support	No Support
Non-Product	86%		14%
Exact	66%	16%	18%
Compatibility	62%	3%	35%
Feature	46%	32%	22%
Thematic	40%	6%	54%
Slang & Symbols	40%	2%	58%
Relational	38%	13%	50%
Product Type	30%	6%	64%
Symptom	24%		76%
Subjective	16%		84%

■ Supported ■ Semi-Support ■ No Support

Source: baymard.com/ecommerce-search

These benchmark results reveal surprisingly dismal support for essential e-commerce search query types. For example, among the top 50 grossing US e-commerce sites:

• 16% of the e-commerce sites do not support that users search by product names (which appear on the product page).

• 18% handle phonetic misspellings so poorly that users will have to pass a spelling test to be presented with results (e.g. 0 results for "Kitchen Aid Artysan" when looking for the "KitchenAid Artisan" mixer).

• 70% require users to search by the exact same product type jargon the site uses, failing to return relevant products for a search such as "blow dryer" if "hair dryer" is used on the site, or "multifunction printer" vs "all-in-one printer".

- 22% of the sites don't support search queries for a color variation (despite the product searched for being available in multiple colors).

- 60% don't support thematic search queries such as "spring jacket" or "office chair".

- 84% don't handle queries that specify a subjective qualifier, such as "cheap" or "high quality".

- 60% don't support symbols and abbreviations, resulting in users missing out on perfectly relevant products if searching for inch when the site has used " or in.

In this article, we'll go over each of 12 query types of e-commerce search – with plenty of query samples, tips on how to best support each query type, and examples from the test sessions.

Deconstructing the Search Query: Spectrum, Qualifiers, and Structure

When users search in an e-commerce context, they are mostly looking for products. This prompts users to search differently than they do when performing generic web searches. In particular, users will often include one or more criteria in their search which the product must meet.

The test subjects would commonly combine the 12 query types when devising their search queries, with certain components of the query "setting the range" of the search while other components are included to refine and delimit that range. Indeed, the 12 query types may be divided into 3 functional groups that map to those intents:

- **Query Spectrum**, *i.e. "setting the search range"* – The query spectrum defines the range of the search, specifying the domain of products that are of interest. It thus provides the base of the search query.

- **Query Qualifiers**, *i.e "delineating the search boundaries"* – Query qualifiers are used to refine the boundaries of the query spectrum by specifying various conditions that the products should meet.

- **Query Structure**, *i.e. "constructing the query"* – The structure of the query determines how it should be interpreted by the search engine and includes the syntax and context of the query.
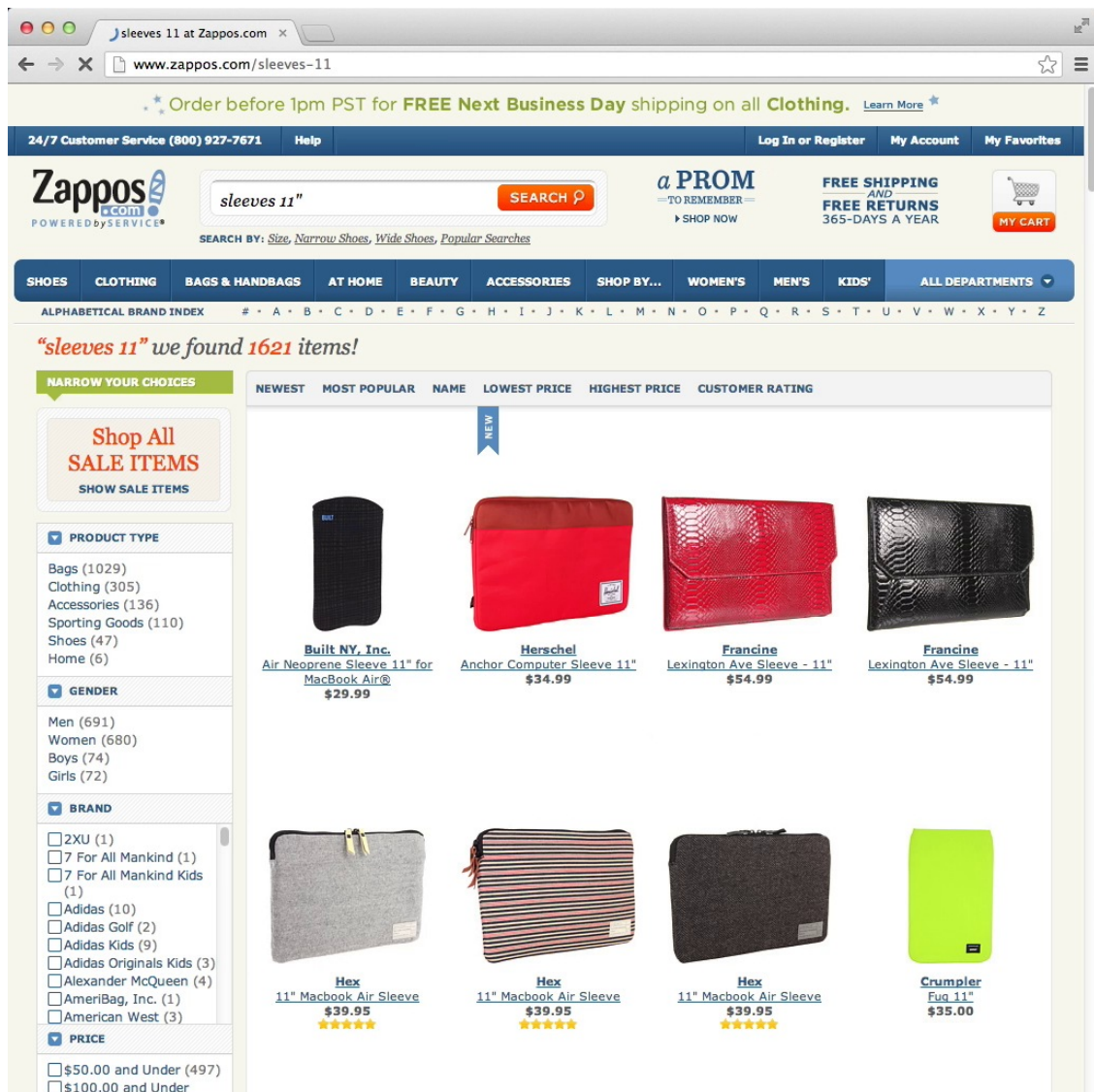
Here one of the test subjects has searched for 'sleeves 11"' when wanting to find a computer bag for her 11" laptop. Notice how 'sleeves' represent the Query Spectrum by specifying the domain of products that should be searched (i.e. laptop sleeves), while '11"' is a Query Qualifier used to refine which laptop sleeves that should be returned (i.e. only ones that fit an 11" laptop). Finally, the Query Structure imply that it is "laptop" sleeves in 11-inch sizes that the user is interested in.

In short, the Query Spectrum is the user's ways of "setting the search range", and they then use Query Qualifiers to "delineate the boundaries" of that range, while the Query Structure determines the syntax and context of the query.

Keep these different functional aspects of the 12 query types in mind as you read through each of them below, as it will help you better understand how users approach search on e-commerce sites. By deconstructing the functional aspect of each query type, its role in the user's query as a whole becomes clearer, which can be immensely helpful when looking to design search logic and interfaces that align with user behavior and expectations.

I) Query Spectrum – "Setting the search range"

Users rely on the query spectrum to indicate the range of what should be searched. Sometimes users want to see a broad collection of products, in which case they will typically set a wide Query Spectrum by, for example, submitting a generic #2 Product Type query such as "Laptops". Other times users are looking for something very specific, and will therefore define a very narrow Query Spectrum, typically in the form of an #1 Exact Search like "Grado Prestige SR325" (a pair headphones).

The query spectrum is the user's way of "setting the search range" and includes 4 query types: #1 Exact Searches, #2 Product Type Searches, #3 Symptom Searches, and #4 Non-Product Searches.

#1 Exact Searches
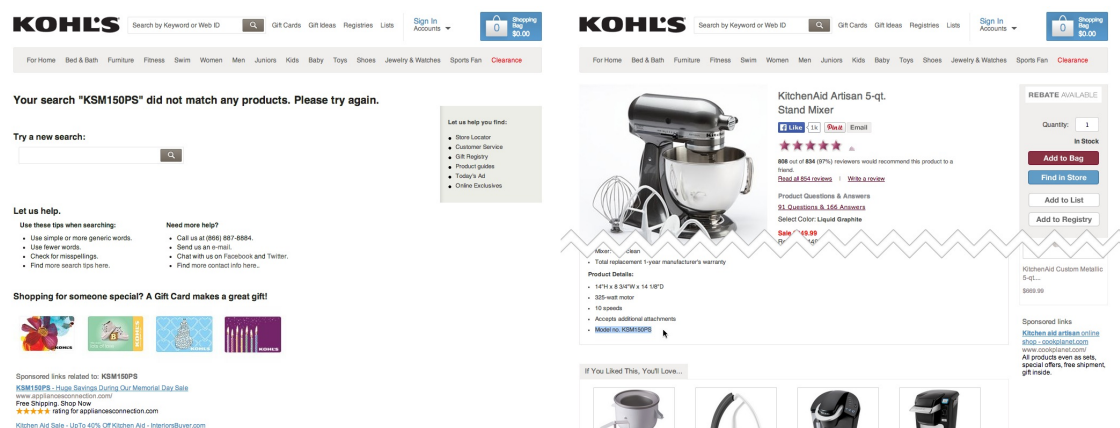
When a user searches for a model name at [Kohl's](#) they won't find anything, even if that exact model number is available in the product description. This is particularly problematic when users copy-paste product titles and model numbers from external sites.

When users know the exact product they are looking for, they will typically rely on #1 Exact Search, entering the product's title or model number, such as "Keurig 45" (a coffee maker).

While this may at first seem like an easy case of keyword matching against those two product attributes, the search engine must be a little smarter than that. For instance, good handling of phonetic misspellings is crucial since the user may only have heard the product title spoken and not know how to spell it, e.g. "Keurick 45".

Furthermore, some products have alternative titles, such as Nestlé Quik vs Nesquik or AT&T Wireless vs Cingular Wireless vs AT&T Mobility – all of which should invoke

their respective product. This is particularly important for global e-commerce sites where products may be localized, such as movies that are called one thing in the US and another in Germany and something entirely else in Argentina.

Another reason it's important to support local and alternate product titles in addition to phonetic misspellings is that users often don't type out the product title themselves. During testing, the subjects were often observed to copy-paste product titles from external sites into the search field. Any misspellings (common when copy-pasted from user-generated content, such as social media posts), or localized or alternate spellings (common when copy-pasted from industry databases, such as IMDb, Goodreads, CarQuery, etc), will therefore be pasted into the search and must therefore be handled gracefully to provide a good search experience.

Acquiring all these local and alternate spellings can prove quite the challenge and vendors may not necessarily input this data on their own. Partnering with relevant industry databases can therefore be a good way to acquire this product information – especially if they are public and users are copy-pasting from them in the first place.

Great support for #1 Exact Searches proved crucial during testing, as the test subjects were observed to quickly conclude that a site didn't carry the product if it didn't show up for a request as specific as a model name or number (as opposed to more open-ended queries, such as #2 Product Type Searches). Yet, of the benchmarked top 50 grossing US e-commerce sites, we found that only 66% are capable of producing decent results for #1 Exact Searches; with 16% being incapable of searching by product names which appear on the product page (like Kohl's example above) and 18% handling misspellings so poorly that Exact queries are significantly hampered (e.g. 0 results for "Kitchen Aid Artysan" when looking for the "KitchenAid Artisan"). In other words, one third of the top 50 grossing US e-commerce sites don't fully support "simple" Exact Searches.

#2 Product Type Searches
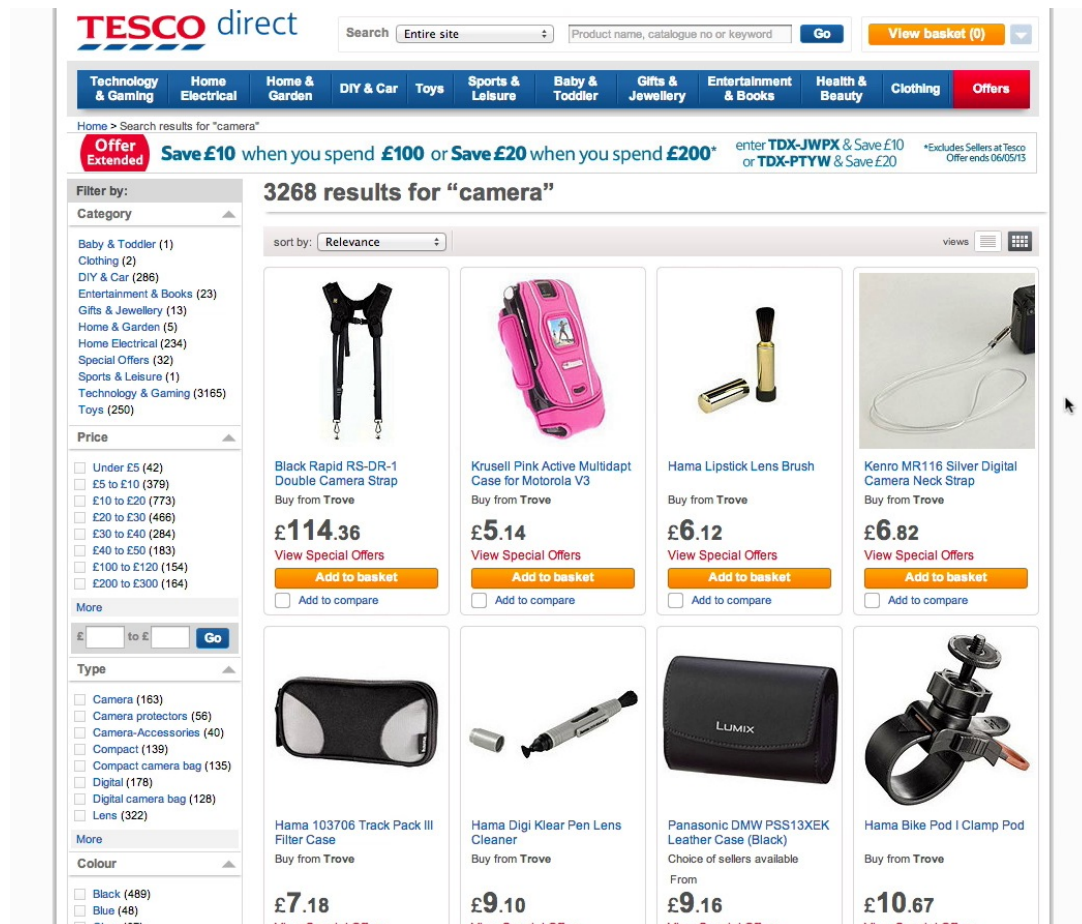
"I had expected some cameras to appear," a test subject muttered after searching for "cameras" on Tesco. She had initially tried finding the category via the site's main navigation, but when she couldn't spot it, she tried searching for the product type instead – but to no avail, as Tesco's search engine clearly doesn't support Product Type searches.

When users aren't looking for a specific product but rather a type of product, they will rely on #2 Product Type Searches, querying for a whole category of products, such as "Sandals". When used on their own, product type searches are generally an attempt by the user to quickly access a category on the site – either because it's more convenient to search for it or because they are having difficulties finding the category on the site.

A very important aspect of supporting #2 Product Type Searches is to return relevant results regardless of whether the searched-for product type exists as a category on the site or not. This not only requires detailed categorization and labelling of products, but also proper handling of synonyms and alternate spellings of those groupings. A search for "t-shirt" should yield the exact same results as one for "tee shirt", regardless of how it happens to be written in each product's title or description. Other examples include "hair dryer" where the user might search for "blow dryer", or users may type "multifunction printer" or even "copy machine" when looking for an "all-in-one printer".

From a user's point of view these everyday descriptions are just as correct as the industry jargon, and most of the test subjects never thought of trying another synonym but instead simply assumed that 0 results for a search such as "copy machine" meant the site didn't carry such products. Despite the severe impact on the user's search experience 70% of the major e-commerce sites do *not* return all the relevant results, if any at all, when users search by a synonym, e.g. "blow dryer" instead of "hair dryer".

Since #2 Product Type Searches are performed by users who are looking to browse a whole category of products, it's crucial that such queries enable relevant filtering and sorting options so the user can easily narrow down the list and compare products. Ideally such filters and sorting options are available directly from the search results (via faceted search filters), although an acceptable alternative is to guide users towards a relevant category scope where such options are available.

#3 Symptom Searches

On Chemist Direct, a test subject searched for "dry cough" only to have hand cream, lip balm, and deodorant returned. Clearly, Chemist Direct's search engine doesn't support Symptom searches and instead returns any item with "dry" in the description.

As we've seen thus far, when users know the specific product they are looking for, they will use Exact search. And if they don't know the exact product or aren't sure about which one they want, they will rely on Product Type searches. However, sometimes users don't even know the type of product they are looking for – all they know is the problem which they are experiencing and that they want a solution to it. In these cases, they will rely on #3 Symptom Searches, entering the problem they are experiencing, such as "stained rug" or "dry cough", in hopes of being presented with viable solutions and products to this problem.

Symptom search is important because it will often be the user's last recourse. If users don't know what solution to look for and can't search for products by entering their problem or symptom, it's going to be awfully difficult for them to find a relevant product on the site. This is especially true if categories on the site aren't problem- or symptom-based (which they often aren't).

Integrating or interlinking any help guides available on the site directly on the search results page is essential too, so that users can learn more about the differences between the different solutions available. Although Symptom searches won't be applicable to every e-commerce industry, they are vital to some, such as drugstores, health & beauty, tools & supplies, cleaning & housekeeping, etc. For sites where Symptom searches are particularly important, it may be a good idea to suggest its usage in the placeholder text of the search field to let users know that they can search by symptom as it isn't all users who think of it.

#4 Non-Product Searches
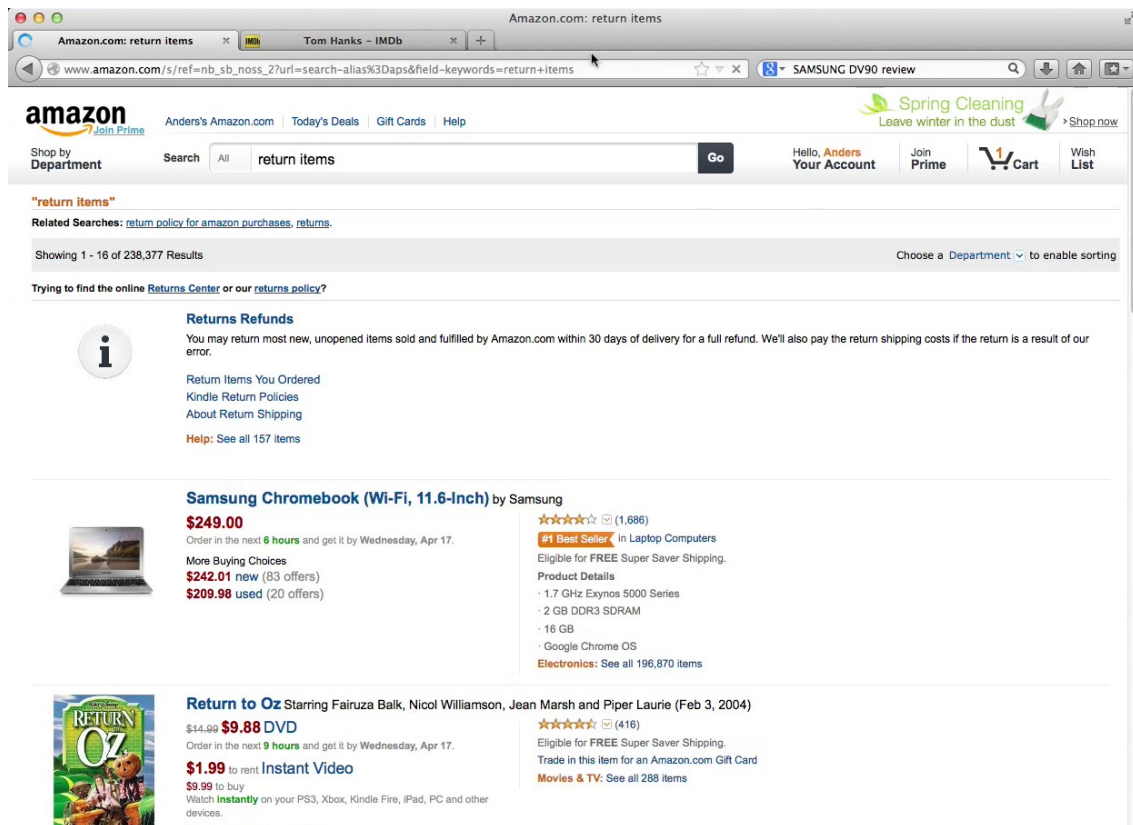
When the test subjects made searches such as "return items" on [Amazon](#), they were greeted by a short one-line description of Amazon's return policy along with a set of links to relevant help sections.

The final type of Query Spectrum is #4 Non-Product Searches, where the user is searching for something that isn't a product, such as the return policy or shipping information. While the primary function of search in an e-commerce context is obviously to find relevant products, the search engine shouldn't be limited to just searching the product catalog, as users will sometimes be looking for other "non-product" content too (help guides, company information, etc).

During testing the subjects would often search for this type of auxiliary content when they had difficulties finding navigational links to it. This is a logical consequence of the content being secondary, and links to it therefore tend to be relegated to the page footer or nested deep within help sections.

Supporting #4 Non-Product Searches enables users to quickly and easily find this information without having to hunt around the site for it. In practice, support for non-product queries is widespread, with 86% of the e-commerce sites either taking the user directly to the relevant page for a search such as "return policy" or presenting the page as a top search result.

'Query Spectrum' Summary

With the Query Spectrum, the user indicates the "range" of what should be searched.

Depending on their particular needs and available information, this can go from highly specific #1 Exact Searches, to broader #2 Product Type Searches, to the inquiry-like #3 Symptom Searches. Finally, users may be looking for non-product pages, in which case they will perform a #4 Non-Product Search.

A summary of the 4 query spectrums can be found in the table below:

I) Query Spectrum – *Used to indicate the range of what should be searched*	
#1. Exact Search *"Keurig K45"*	Searching for specific products by title
#2. Product Type Search *"Sandals"*	Searching for groups or whole categories of products
#3. Symptom Search *"Stained rug"*	Searching for products by querying for the problem they must solve
#4. Non-Product Search *"Return policy"*	Searching for help pages, company information, and other non-product pages

II) Query Qualifiers – "Delineating the search boundaries"

Users rely on query qualifiers to refine and delimit the boundaries of the Query Spectrum. These qualifiers are typically used when the user has a set of criteria that they want the product to meet. For example, the user may try to limit the type of TVs returned by submitting a #5 Feature query such as "Plasma TV" or perform a #8 Compatibility search like "Lens for Nikon D7000" to filter out any incompatible lenses.

Query qualifiers are the user's way of "delineating the search boundaries" and includes 5 query types: #5 Feature Searches, #6 Thematic Searches, #7 Relational Searches, #8 Compatibility Searches, and #9 Subjective Searches.

#5 Feature Searches

On [Hayneedle](#), searching for "manual espresso machine" sends the user to an "Espresso Machines" category with a "Manual Espresso Machines" filter applied, resulting in a list of highly relevant search results.

Users often have a set of criteria that they want the product to meet, and very often these criteria relate to features of the product. #5 Feature Searches is when the user includes one or more features in their search query that they want the product to have. For example, a user may not want just any "jacket" but will often be looking for something slightly more specific such as a "leather jacket", or a "red vase" rather than simply a "vase".

The term "Features" should be understood in a rather broad sense referring to any type of product aspect or attribute. Features can thus be the product's color ("red dresses"), material ("fabric sofas"), performance specs ("100000 IOPS hard drive"), or format ("Hobbit DVD"), not to mention price ("$100-$200 backpacks") or brand ("Nike running shoes"). The list goes on, and all significant attributes should be searchable, even if they don't apply to all products on the site. For example, even if all products on the site don't have a "format" attribute, movies on the site should still be searchable by it.

During testing, #5 Feature Searches were the by far most common Query Qualifier, making it a definite "must have" of the 12 Query Types. Among the top e-commerce sites 46% support that users search by a wide range of features (e.g. color, material, brand), while 32% only have proper support for color searches, with the remaining 22% not even supporting color searches (for the products in multiple color variations).

In terms of implementation, the ideal solution is to dynamically apply the features searched for as filters on the results page, as this increases transparency and user control – with the user being able to see what is and isn't included and being able to quickly toggle related filters on/off. For instance, in the above Hayneedle example, the "Manual" aspect of the "Manual Espresso Machine" query is applied as a filter, with the user being able to see and toggle the other available types of espresso machines (semi-automatic, stove top, etc).

#6 Thematic Searches
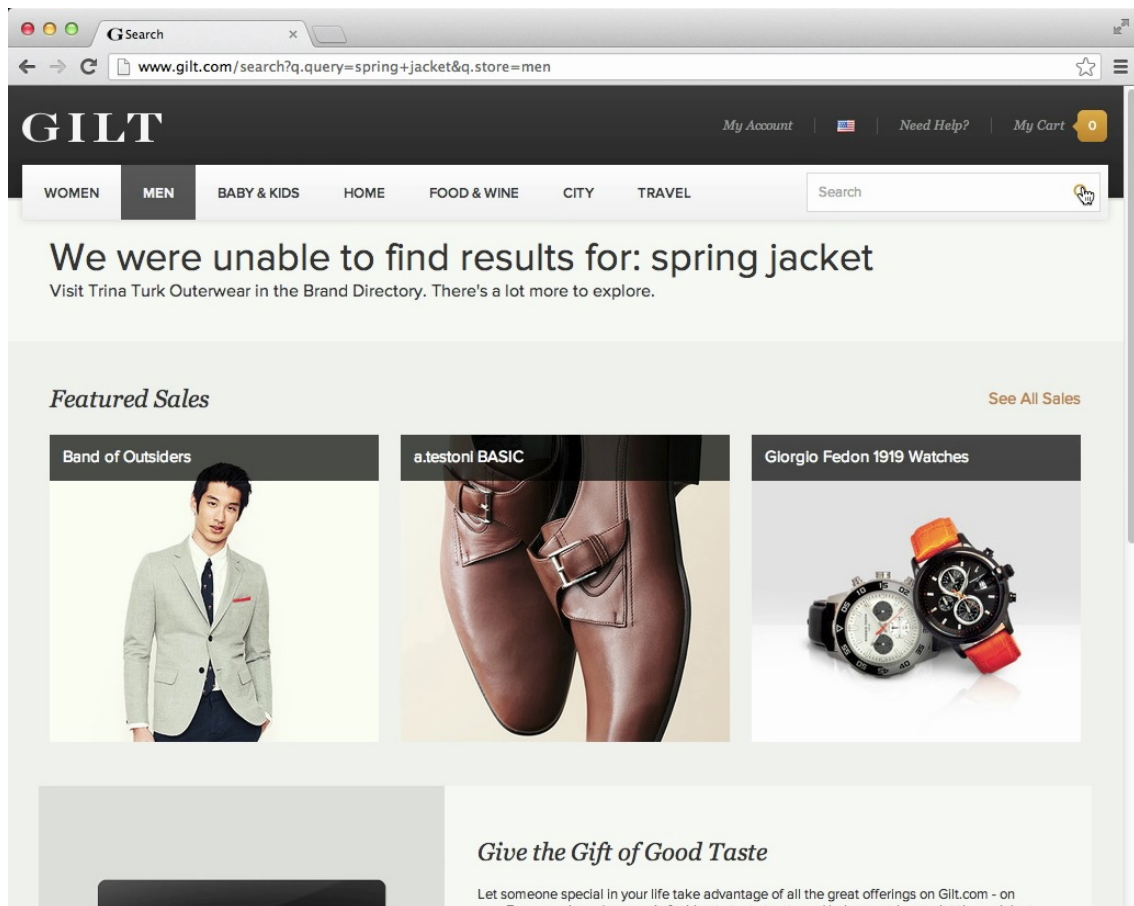
A test subject searching for "spring jacket" on [Gilt](https://www.gilt.com), was awarded with zero results, jovially commenting, "That wasn't exactly impressive. Haha. It didn't give me a damn thing."

What exactly constitutes a "living room rug" or an "extreme weather sleeping bag"? While these are certainly concepts we can easily recognize, defining their exact boundaries can be a challenge. #6 Thematic Searches are often a little difficult to define because they are inherently vague in nature – they often include fuzzy boundaries (e.g., "living room") or categories of intended usage (e.g., "spring", "cold weather"). Nonetheless, they are very real notions to users who – in certain industries such as apparel – include thematic qualifiers liberally in their searches.

A great deal of interpretation is generally required to support Thematic searches, both in terms of the meaning of the actual query itself but also in the internal tagging of products, as it's vital that a query for e.g. "spring jacket" presents all the relevant products, not just the handful of products which happen to have those keywords in their title or description. This will often require some sort of thematic tagging of the product catalog to determine, e.g. which jackets would be suitable for spring (and which wouldn't). Meanwhile, a search for a "Mother's Day bouquet" requires products to be tagged with an occasion (a social concept). Despite thematic product browsing being a common browsing

pattern and product arrangement in physical retail, 60% of the top e-commerce sites do not support thematic search queries.

Typical Thematic searches include seasons (e.g., "spring"), intended usage (e.g., "office" or "outdoors"), occasions (e.g., "birthday" or "wedding"), events (e.g., "Olympics" or "NBA"), etc. Unsurprisingly, these more or less map to thematic categories, and they may therefore even have their own custom category pages. This is yet another case where good product categorization and labelling will get you half the way there.

#7 Relational Searches

Here a test subject has searched for "Steven Spielberg" to get a list of the movies he's been involved with – luckily, [Amazon](#) supports relational searches and were able to present the user with a highly relevant list of movies by the prolific movieman. (Notice how the user is also presented with a list of related directors to switch between in the sidebar.)

Sometimes users care more about the people affiliated with a product than the product itself. This is especially true of entertainment products such as books, music and movies, although it extends beyond these to other domains such designer items (furniture, jewel-

ry, etc). In these cases, users will often rely on #7 Relational Searches, where they enter the name of entities involved with or related to the product.

In some cases, users will also add an additional "role" qualifier to their #7 Relational Search, specifying the nature of the entity's relationship to the product. For example, searching for "James Franco" could meaningfully bring up the books he's authored, the blockbuster movies he's acted in, or the art films he's directed – meanwhile, the user likely had only one of those product types in mind.

#7 Relational Search is the Query Qualifier that is most commonly submitted as a standalone query. However, this is typically because the person is only known for one thing (or the user only knows them for one body of work) and the Query Spectrum is therefore #11 Implicit – the user takes it for granted that movies are going to show up when they search for "Tom Hanks".

#8 Compatibility Searches

Here a test subject has searched for "charger lenovo ideapad yoga" in hope of finding a new power adapter for her Lenovo IdeaPad Yoga laptop. Unfortunately, [Best Buy](#) doesn't sup-

port compatibility searches, and so instead of returning "chargers" (i.e. power adapters) the site returns a bunch of Lenovo IdeaPad laptops.

Users often don't know the name of the accessory or spare part they need – instead they know the details of the product they already own. It's therefore not uncommon to see users perform #8 Compatibility Searches where they input the name or brand of a product they own along with the type of accessory or spare part they are looking for, such as "sony cybershot camera case".

Compatibility search is similar to #7 Relational Searches, except it requires strict compliance – it's two or more products that must work together, with the user trying to find products that are specifically compatible with a product they already own (or are about to buy).

Users who know the exact model of their product will typically submit this, and #8 Compatibility Searches should therefore obviously work with product models. In fact, sometimes compatibility can only be ensured if the exact product model is provided. However, users don't always know or recall their specific product model, and support for more generic product types and brand names is therefore important as well, so that users can enter generic queries like "13in laptop sleeve" rather than specifying the exact laptop they own. Generic compatibility queries are among the most supported query types, with 62% of the sites that sell compatibility items supporting basic compatibility queries.

Integration with product finders and help content is often a good idea, especially for more generic queries that don't include sufficient information to fully determine compatibility.

#9 Subjective Searches

Like 84% of the major e-commerce sites, Wayfair doesn't support subjective qualifiers, as is evident from this search for "high-quality kettle" which returns items with either no or poor ratings. This is despite Wayfair having plenty of kettles with 4.5+ star averages and numerous reviews, and while the first two results aren't cheap, they are far from Wayfair's most expensive kettles.

What exactly constitutes a "high-quality" product? Or a "nice-looking" or "cheap" one? Answers to such questions will necessarily be subjective in nature, yet these are exactly the types of qualifiers that some users have in mind when looking for products – and it's therefore not uncommon for users to perform such #9 Subjective Searches.

When dealing with subjective qualifiers, the site must often try to approximate the user's intent by using one or more attributes as a proxy for the subjective term(s). For example, if presented with the earlier #9 Subjective Search for "high-quality kettle", one could reasonably favor products with high scores across a wide range of product attributes, such as user reviews, number of sales, and price, and use those to calculate the most relevant matches. Similarly, if a user wants to find a "cheap wine", one could compute the total price range for "wine" and then select the bottom 15% (i.e., the cheapest) of those bottles. Or the search could simply return the "wine" category sorted by price.

Taste-based #9 Subjective Searches are by far the trickiest to program a response to. How do you programmatically identify "nice-looking" products? While, in some cases, there won't be any good ways to answer such questions, there are some valid solutions to certain types of taste-based queries. For example, if a user searches for "beautiful tables" on a furniture site, it's true that "beautiful" may be difficult to pinpoint, but as a response the user could be asked to select from different styles of tables available (modern, antiques, glass, Asian, etc). In effect, asking the user to self-select what "beautiful" means to them.

A good way to identify an attribute that may serve as a useful proxy for a given taste-preference is to deconstruct what the taste is really about. In the case of "beautiful", we're dealing with aesthetics, and any product attributes related to the product's aesthetic may therefore be viable proxies. We can see how this works for "delicious" too. Say a user searches for "delicious snacks". While "delicious" is very subjective to each individual user, it has to do with flavor, and the user could therefore be asked to identify the type of flavor they are interested in (sweet, savory, salty, etc).

Despite subjective qualifiers often being vital to the user's purchase decision the vast majority of the top grossing e-commerce sites (84% to be exact) don't handle queries that specify even a basic subjective qualifier such as "cheap".

'Query Qualifiers' Summary

With Query Qualifiers, the user refines the boundaries of the Query Spectrum range by including various conditions that the products must meet.

Query Qualifiers almost never make sense as standalone queries and therefore tends to be combined with a Query Spectrum (with the possible exceptions being #6 Thematic and #7 Relational searches).

Depending on the user's approach and knowledge about the product domain, they may specify certain #5 Features that the product must have, or describe a #6 Thematic context in which it must be used. Users may similarly try to find products by their #7 Relation to other objects, or define technical #8 Compatibility requirements. Finally, some users will inject #9 Subjective criteria that must be met (or approximated).

While especially thematic, compatibility and subjective searches are somewhat "advanced" query types from a technical perspective, they are trivial requests in physical retail ("cheap spring jacket" or "nice leather case for my Nikon camera") and were used frequently by the subjects during testing.

A summary of the 5 query qualifiers can be found in the table below:

II) Query Qualifiers – *Used to delimit the Query Spectrum, specifying conditions for what should and/or shouldn't be included*	
#5. Feature Search *"Waterproof cameras"*	Searching for products with specific attributes or features
#6. Thematic Search *"Living room rug"*	Searching for categories or concepts that are vague in nature or have "fuzzy" boundaries
#7. Relational Search *"Movies starring Tom Hanks"*	Searching for products by their affiliation with another object
#8. Compatibility Search *"Lenses for Nikon D7000"*	Searching for products by their compatibility with another item
#9. Subjective Search *"High-quality kettles"*	Searching for products using non-objective qualifiers

III) Query Structure – "Constructing the query"

The structure of the query determines how it should be interpreted by the search engine. It involves both the linguistic syntax of the query and the circumstances under which it was conceived.

For instance, users may take certain aspects of their search for granted and leave them out of their query, resulting in #11 Implicit Searches such as "Pants" when what the user is really interested in is "Women's Pants". The syntax of the user's query can be highly important too, with a symbol such as '-' taking on significantly different meanings and functions in #10 Slang, Abbreviation, and Symbol Searches like "$10-100 sleeping bags" and "-10 deg. sleeping bags".

The query structure deals with how the query is constructed – the context in which it was written and the syntax of the query itself – and includes 3 query types: #10 Slang, Abbreviation, and Symbol Searches, #11 Implicit Searches, and #12 Natural Language Searches.

#10 Slang, Abbreviation, and Symbol Searches
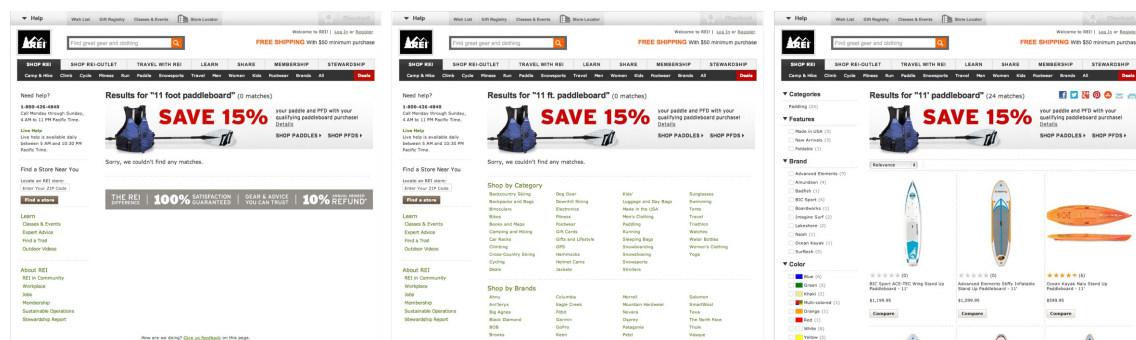
On [REI](), searches for "11 foot paddleboard" and "11 ft. paddleboard" yield 0 results while "11' paddleboard" returns 24. Due to this lack of support for slang, abbreviation, and symbol searches, it's only the few lucky users who by chance happen to use the exact same spelling as the site that will get any results when they search.

Users rely on a wide range of linguistic shortcuts when they search. This quickly became evident during testing as several of the subjects frequently relied on #10 Slang, Abbreviation, and Symbol Searches, writing slang-based queries such as "RayBan shades", or including abbreviations like "13in laptop sleeve", or relying on symbols such as "sleeping bag -5 degrees".

Slang and abbreviations are by far the easiest to support technically as it essentially just requires mapping between different terms, pairing slang words like "kicks" with "shoes" and "fixie" with "fixed-gear bike"; similarly abbreviations must be mapped so "ml" pairs up with "millilitre" and "HP" with "Hewlett-Packard".

Symbols can prove a little tricker since they may act as more than mere synonyms and may change meaning depending on the word arrangement of the query. For example, the "-" symbol could be used to denote both a minus (e.g., "sleeping bag for -10 deg.") and a range (e.g., "sweaters $50-$100"). In these instances, not only does the meaning of the symbol change, but the symbol also acts as a filtering instruction.

When benchmarking the top 50 sites, we tested perhaps the most simple size-related symbol search possible – *inch* versus the ″ character – yet a whopping 60% of the e-commerce sites showed different results depending on whether a symbol, text or abbreviation was used. In other words, on 60% of the top grossing US e-commerce sites, users will miss out on perfectly relevant products because they search for either *inch*, ″ or *in*.

#11 Implicit Searches

Here a test subject simply searched for "adventure" without including "action figure", even though it was this particular type of toy she was looking for. The Entertainer could have

inferred this implicit component of the user's query from the category page where she conducted the search as well as the many previous product pages she had visited during this session on the site.

From time to time, users will leave out certain aspects of their search query. The omission is often not a conscious decision, but rather happens because the user takes the aspect for granted due to their current context or frame of mind. Dealing with such #11 Implicit Searches requires the site to make use of all available environmental data in order to accurately infer any implied aspects of the user's query.

The most obvious variable is the page the user searches from – for example, if a user is in a "Women's Dresses" category and searches for "pants", it is reasonable to assume that the user is looking for women's pants and not men's or children's pants. Other such variables include past pages the user has visited on the site, products in the user's shopping cart, demographic information (either approximated or from any available account data), purchase history, how the user entered the site, duration since last visit, duration of current visit, and so on. Of course, if the user has an account and is signed in, information from their profile may also be used to inform #11 Implicit Searches.

Once an implied component has been detected, there are a few ways the search experience may be altered. One approach is to bias the search results towards the implied aspects – following our earlier example, the site may bias pants from the women's scope when a search is performed from one of the women's clothing categories. This makes for a very seamless user experience although it lacks transparency. Alternatively, continuing our women's pants example, the results page may suggest relevant search scopes or query clarifications, such as *"Did you mean 'women's pants'?"*

Finally, in the case of a very high confidence in the user's intent, the site may go as far as auto-refining the user's search query – that is, the site auto-corrects the user's query to include the implied component(s). In this scenario, it's extremely important to state the change very clearly to the user and offer them a way to "force" through their original query, so the auto-correction doesn't become restrictive.

#12 Natural Language Searches

Introducing Graph Search

Try Graph Search

🔍 Photos of my friends in **New York**|

Explore your world through photos

Now you can use simple, specific phrases like "Photos my friends took in New York City" to find anything you want.
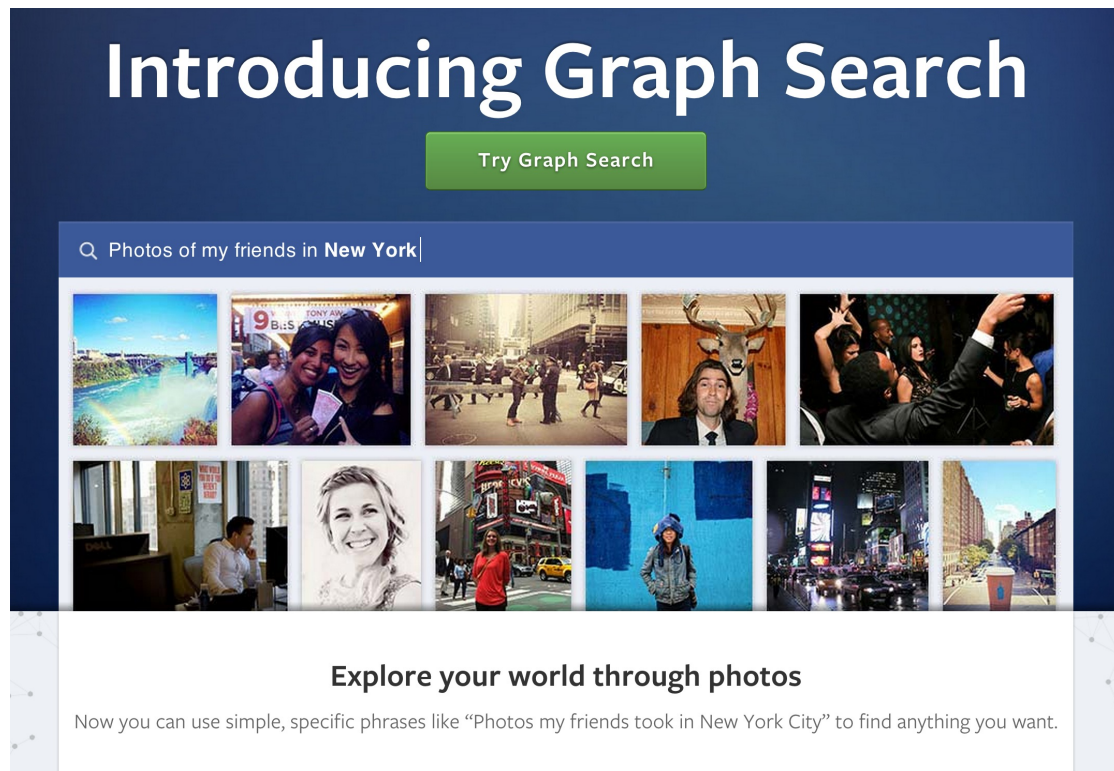
Being able to enter search queries using everyday sentences, such as "Photos of my friends in New York", give users an incredibly simple and straight-forward way to perform highly advanced searches.

#12 Natural Language Search is when a user submits a query that uses regular spoken language. Ideally, the search engine is able to interpret the meaning of this query and return highly relevant results, going well beyond simple keyword matching.

At its core, Natural Language search is about understanding the semantics, context, and relationships of the query, rather than approaching the search query as a simple set of keywords. When done right, this allows the user to articulate a question or request, such as "women's shoes that are red and available in size 7.5", and ideally this returns a list of red women's shoes in size 7.5.

What's striking here is how complex that query actually is when you break it down – it includes filters for the category ("women"), product type ("shoes"), and two types of product variations ("red" and "size 7.5"). Normally, applying all these filters would require a fairly advanced interface and lots of clicking on part of the user – yet with #12 Natural Language Search, the interface is as basic as it gets: just say what you want and you'll get it. (In fact, with the increasing support for voice input the query may literally be spoken.)

While Natural Language searches are still far from being commonplace, recent years have shown increasing application. Google's search, Apple's Siri, and Facebook's Graph Search, are perhaps some of the most well-known cases. These companies have upped the ante and can properly interpret a wide range of Natural Language queries and offer the user a meaningful response in return.

'Query Structure' Summary:

The Query Structure deals with how the user has put together their query, and consequently how it should be interpreted by the search engine. It's about the syntax of the query and the context in which it was written. For instance, the user may apply various linguistic shortcuts in the form of #10 Slang, Abbreviations, and Symbols when they devise their search query. Other times, users may leave certain parts of their query #11 Implicit, typically because they take it for granted given their current context.

Finally, as online search matures and voice input proliferates, #12 Natural Language searches will grow increasingly common – requiring search engines to take on a new level of linguistic intelligence but potentially also helping e-commerce sites approximate the level of support offered in physical retail outlets by human salespeople.

A summary of the 3 query structures can be found in the table below:

III) Query Structure – *How the query is constructed by the user and interpreted by the search engine*	
#10. Slang, Abbreviation, and Symbol Search *"Sleeping bag -10 deg."*	Searching for products using various linguistic shortcuts
#11. Implicit Search *"[Women's] Pants"*	Forgetting to include certain qualifiers in the search query due to one's current frame of mind
#12. Natural Language Search *"Women's shoes that are red and available in size 7.5"*	Searching in full sentences rather than bundles of keywords

Improving Support for the 12 Query Types

After testing 19 major sites and auditing the search experience of the 50 top grossing US e-commerce sites, we can confidently say that support for the 12 query types presented in this article is lackluster at best.

Let's take another look at the graph presented earlier to see the level of support for each query type. (Note: #11 Implicit and #12 Natural Language Search were omitted as support for these query types couldn't be reliably tested across different sites and industries.)

Search Query Support Among Top 50 US E-Commerce Sites

Query Type	Supported	Semi-Support	No Support
Non-Product	86%		14%
Exact	66%	16%	18%
Compatibility	62%	3%	35%
Feature	46%	32%	22%
Thematic	40%	6%	54%
Slang & Symbols	40%	2%	58%
Relational	38%	13%	50%
Product Type	30%	6%	64%
Symptom	24%		76%
Subjective	16%		84%

■ Supported ■ Semi-Support ■ No Support   Source: baymard.com/ecommerce-search
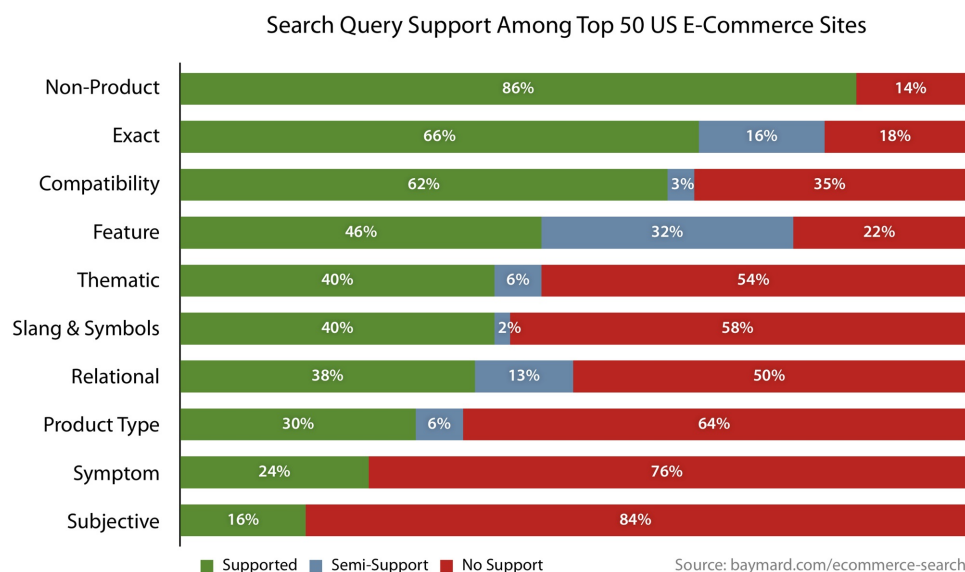
In the free, publicly available part of the [E-Commerce Search Usability benchmark](https://baymard.com/ecommerce-search) database you can also see how each of the 50 major e-commerce sites ranks when benchmarked on their entire search experience (and not just their query support).

When e-commerce sites with millions of dollars in revenue show this low level of support for essential query types such as #2 Product Type and #7 Relational searches, you know the current state of e-commerce search is dire. And since e-commerce search engines are nearly always reused across platforms (different interfaces accessing the same search engine API), query support will be similarly poor on tablet and mobile sites / apps too.

The good news is that this means plenty of opportunity to rise above the competition. Creating a (comparatively-speaking) superior search experience only requires proper support of 6-7 query types. In fact, supporting the right handful of the most essential query types is enough to create a decent search experience. We recommend that you start out by making sure that your search engine supports these 5 essential query types: #1 Exact, #2 Product Type, #5 Feature, #6 Thematic, and #7 Relational Searches (where applicable). Users can get by with basic e-commerce searches when these 5 query types are support. Conversely, failing to support any of these core query types will result in a defective search experience.

To create a truly great search experience, 8-10 of the query types must be supported. This is not done overnight and often requires more than simply investing in good search logic – detailed and structured product data is often just as important. The good news is that the investments can pay significant dividends as users begin finding (and thus are able to buy) the products they are looking for. Also, since the same search engine tends to be accessed by all platforms, any investments will typically pay off universally across those platforms – improving the search experience of your e-commerce desktop website and native mobile apps alike.

During testing we observed how the test subjects were greatly influenced by prior experience with the site. If they had previously had success searching for something on the site, they were much more likely to use search on that site even if they generally preferred category navigation. And vice-versa: prior poor search experiences steered otherwise search-happy subjects towards category navigation. Thus, not investing in good search usability can not only cost sales in the short- and mid-term, it can also set flawed user expectations for future use of your site. Expectations which can be difficult to shake off even if the search experience is eventually improved down the road.

We therefore urge e-commerce owners not to post-pone investing in improvements to their e-commerce search experience, lest they be haunted by faulty user perceptions for years to come. Instead, take advantage of the dismal state of e-commerce search and turn it into a competitive advantage.

It's time to improve the state of e-commerce search – and that starts by improving your search engine logic with support for the 12 query types:

I) Query Spectrum – *Used to indicate the range of what should be searched*		
Query Type	*User behavior*	*How you can support it*
#1. Exact Search *"Keurig K45"*	Searching for specific products by title	Basic keyword matching, along with support for multiple title variations and intelligent handling of misspellings
#2. Product Type Search *"Sandals"*	Searching for groups or whole categories of products	Support for synonyms as well as categories that aren't part of the site's navigation / hierarchy
#3. Symptom Search *"Stained rug"*	Searching for products by querying for the problem they must solve	Symptom database mapping "symptoms" to "cures" (i.e. problems to solutions)
#4. Non-Product Search *"Return policy"*	Searching for help pages, company information, and other non-product pages	Search engine must index the entire website, not just products

II) Query Qualifiers – *Used to delimit the Query Spectrum, specifying conditions for what should and/or shouldn't be included*		
Query Type	*User behavior*	*How you can support it*
#5. Feature Search *"Waterproof cameras"*	Searching for products with specific attributes or features	Intelligent parsing of product specifications (i.e. structured product data)
#6. Thematic Search *"Living room rug"*	Searching for categories or concepts that are vague in nature or have "fuzzy" boundaries	Interpretive labelling of products and categories
#7. Relational Search *"Movies starring Tom Hanks"*	Searching for products by their affiliation with another object	Association data linking products and objects, ideally specifying the nature of the relationship too
#8. Compatibility Search *"Lenses for Nikon D7000"*	Searching for products by their compatibility with another item	Compatibility database mapping compatible products to one another
#9. Subjective Search *"High-quality kettles"*	Searching for products using non-objective qualifiers	Handling of quantifiable single-attribute degrees (e.g. "cheap"), quantifiable but multi-attribute mix ("value for money"), and tasted-based ("delicious") qualifiers

III) Query Structure – *How the query is constructed by the user and interpreted by the search engine*		
Query Type	*User behavior*	*How you can support it*
#10. Slang, Abbreviation, and Symbol Search *"Sleeping bag -10 deg."*	Searching for products using various linguistic shortcuts	Synonym mapping of slangs, abbreviations, and symbols, as well as interpretation of symbol intent (ranges, modifiers, etc)
#11. Implicit Search *"[Women's] Pants"*	Forgetting to include certain qualifiers in the search query due to one's current frame of mind	All available environmental variables must be used to infer any implicit aspects of the user's query
#12. Natural Language Search *"Women's shoes that are red and available in size 7.5"*	Searching in full sentences rather than bundles of keywords	Intelligent parsing and deconstruction of the user's query