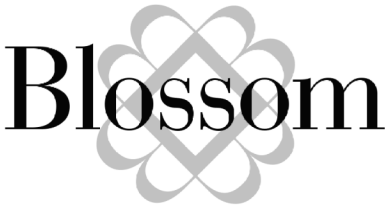

What are Phrasal Query Suggestions?

An advanced feature of Blossom Search is its suggestion feature for modifying a query. When a query returns too few results, Blossom Search suggests more general queries that are *guaranteed* to return more results. Similarly, when a query returns too many results, the search engine suggests more specific queries *guaranteed* to return fewer results. We call these *phrasal query* suggestions because both are based on treating a query as a set of related terms, i.e., a phrase.

Read on for more details on how suggestions are generated.

The appeal of proximity search

It is easy to see that precise meaning is nearly always expressed in phrases rather than individual words. This is largely because most words refer to general concepts, so we use phrases to choose narrow the meaning. For example, there are lots of types of "engine", so a phrase such as "search engine" chooses a specific type.

When a search engine implements *proximity search*, all of the terms in a query must appear near one another, usually within a small fixed-size window or perhaps within a semantic structure like a sentence. Requiring the terms to be in the same context improves the accuracy of searching–we say proximity search has high precision. The downside is that proximity search may miss some relevant documents–it is said to have low recall.

Usual solutions to improve recall

There are two common ways search engines improve the recall of proximity search: relax the proximity constraint and relax the all-terms constraint. Unfortunately, both approaches increase the number of irrelevant documents in the search results.

A *conjunctive* search is like a proximity search, but without the requirement that all the terms appear near one another. In a conjunctive search, if all the query terms appear *anywhere* within a document, then it is considered relevant to the query. Conjunctive search is sometimes called an AND search; the results can be represented by the intersection of the documents containing each of the search terms (please see Figure 1). It

is easy to see that conjunctive search will return all the same documents as proximity search, plus those where the terms aren't near one another. Unfortunately, on average those additional documents won't be as relevant as those that meet the proximity constraint.

In a *disjunctive* search, a document is considered relevant if it contains *any* of the terms in the query. Disjunctive search is sometimes referred to as an OR search, and its results can be represented by the union of the documents containing the search terms (see Figure 2). Again, it is easy to see that disjunctive search will return the same documents as proximity and conjunctive search, plus others. And again, it is easy to see the additional documents, on average, will be less relevant than those that meet the stricter constraints.

In summary, top ranked documents are usually those that meet the proximity-search condition. To improve recall, most search engines add documents that meet weaker constraints (and are thus, on average, less relevant). Instead of using weaker constraints, Blossom Search offers query suggestions.

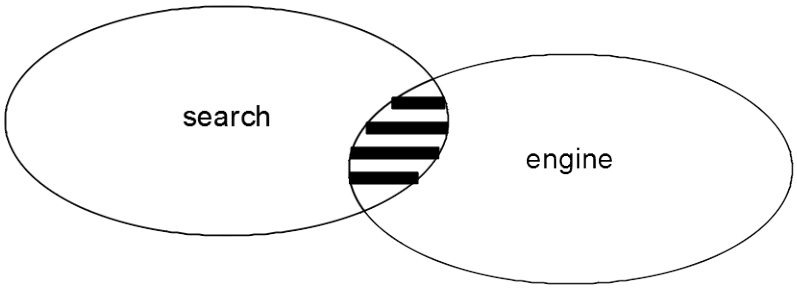

Figure 1. Conjunctive search can be represented as the intersection of sets. For the query "search engine" relevant documents are those that occur in the intersection of the set of documents that contain "search" and the set that contains "engine".

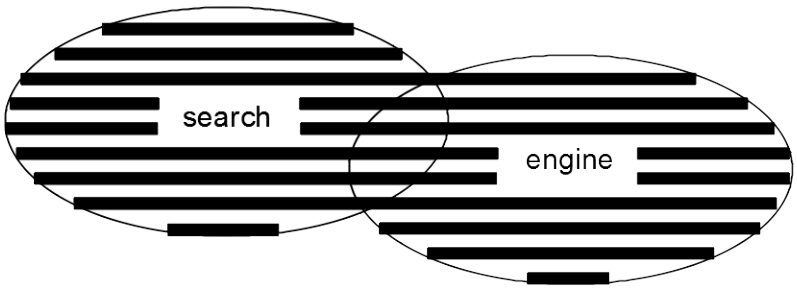

Figure 2. Disjunctive search can be represented as the union of sets.

Query Suggestions

If a query returns no results, it is too narrow–it needs to be broadened. Removing query terms broadens a proximity search query. For example, the query “search engine” is broader than “Blossom’s search engine” and will match more documents. Similarly, if a query returns too many results, it is probably too broad. Adding query terms narrows a proximity search, thus “Blossom’s search engine” is narrower than “search engine” and will match fewer documents.

Unfortunately, disjunctive search does not share these intuitive properties of proximity (and conjunctive) search. Adding a term *increases* the number of matched documents, so “Blossom’s search engine” is actually *broader* than just “search engine”.

Generalization: drop terms

The Blossom search engine utilizes the intuitive properties of proximity search to generate phrasal suggestions. When a query generates too few results, the engine suggests more general queries that are subphrases of the original query. It creates suggestions by performing a proximity search for every possible subphrase, ranking them based on their length and number of occurrences. Each suggestion is thus guaranteed to return more results.

Specialization: add terms

To specialize a query that returns too many results, Blossom suggests adding terms to the original query. The terms suggested come from extracting all of phrases in all of the documents covered by the search index. Phrases that contain all the query terms are ranked by their length and number of occurrences. Again, each suggestion is guaranteed to return fewer results than the original query.