

Full-Text Searching Help

Boolean Searching

A boolean search request consists of a group of words or phrases linked by connectors.

apple and pear	Both words must be present.
apple or pear	Either word can be present.
apple w/5 pear	<i>Apple</i> must occur within 5 words of <i>pear</i> .
apple and not pear	Only <i>apple</i> must be present.
apple not w/5 pear	<i>Apple</i> must not occur within 5 words of <i>pear</i> .

If you use more than one connector, you should use parentheses to indicate precisely what you want to search for. For example, *apple and pear or orange juice* could mean *(apple and pear) or orange*, or it could mean *apple and (pear or orange)*.

AND Connector

Use the AND connector in a search request to connect two expressions, both of which must be found in any document retrieved. For example:

- *apple pie* and *poached pear* would retrieve any document that contained both phrases.
- *(apple or banana)* and *(pear w/5 grape)* would retrieve any document that (1) contained either *apple* OR *banana*, AND (2) contained *pear* within 5 words of *grape*.

OR Connector

Use the OR connector in a search request to connect two expressions, at least one of which must be found in any document retrieved. For example, *apple pie or poached pear* would retrieve any document that contained *apple pie*, *poached pear*, or both.

Proximity Searching W/number

Use the *W/number* connector in a search request to specify that one word or phrase must occur within a certain number of words of the other. For example, *apple w/5 pear* would retrieve any document that contained *apple* within 5 words of *pear*. The following are examples of search requests using *W/number*:

- *(apple or pear) w/5 banana*
- *(apple w/5 banana) w/10 pear*
- *(apple and banana) w/10 pear*

Some types of complex expressions using the *W/number* connector will produce ambiguous results and should not be used. The following are examples of ambiguous search requests:

- *(apple and banana) w/10 (pear and grape)*
- *(apple w/10 banana) w/10 (pear and grape)*

In general, at least one of the two expressions connected by *W/number* must be a single word or phrase or a group of words and phrases connected by OR. Example:

- (apple and banana) w/10 (pear or grape)
- (apple and banana) w/10 orange tree

This system uses two built in search words to mark the beginning and end of a file: *xfirstword* and *xlastword*. The terms are useful if you want to limit a search to the beginning or end of a file. For example, *apple w/10 xlastword* would search for *apple* within 10 words of the end of a document.

NOT and NOT W/number

Use NOT in front of any search expression to reverse its meaning. This allows you to exclude documents from a search. Example:

- apple sauce and not pear

NOT standing alone can be the start of a search request. For example, *not pear* would retrieve all documents that did not contain *pear*.

If NOT is not the first connector in a request, you need to use either AND or OR with NOT:

- apple or not pear
- not (apple w/5 pear)

The NOT W/ ("not within") operator allows you to search for a word or phrase not in association with another word or phrase. Example:

- apple not w/20 pear

Wildcards and Special Searches

?	Wildcard replacing single character. Example: <i>appl?</i> matches <i>apply</i> or <i>apple</i> .
*	Wildcard replacing any number of characters. Example: <i>appl*</i> matches <i>application</i> .
~	Stemming. Example: <i>apply~</i> matches <i>apply</i> , <i>applies</i> , <i>applied</i> .
%	Fuzzy Search. Example: <i>ba%nana</i> matches <i>banana</i> , <i>bananna</i> .
#	Phonic Search. Example: <i>#smith</i> matches <i>smith</i> , <i>smythe</i> .
&	Synonym Search. Example: <i>fast&</i> matches <i>quick</i> .
~~	Numeric Range. Example: <i>12~~24</i> matches <i>18</i> .

Wildcards (* and ?)

A search word can contain the wildcard characters * and ?. A ? in a word matches any single character, and a * matches any number of characters. The wildcard characters can be in any position in a word. For example:

- *appl** would match apple, application, etc.
- **cipl** would match principle, participle, etc.
- *appl?* would match apply and apple but not apples.
- *ap*ed* would match applied, approved, etc.

Use of the * wildcard character near the beginning of a word will slow searches.

Synonym Searching (&)

Synonym searching finds synonyms of a word in a search request. For example, a search for *fast* would also find *quick*. You can enable synonym searching for all words in a request or you can enable synonym searching selectively by adding the & character after certain words in a request. Example: *fast& w/5 search*.

Fuzzy Searching (pull-down menu, %)

Fuzzy searching will find a word even if it is misspelled. For example, a fuzzy search for *apple* will find *appple*. Fuzzy searching can be useful when you are searching text that may contain typographical errors, or for text that has been scanned using optical character recognition (OCR).

- You can set the degree of fuzziness using the pull-down menu. *Very Fuzzy* will return more hits than *Fuzzy*.
- You can add fuzziness selectively using the % character. The number of % characters you add determines the number of differences the system will ignore when searching for a word. The position of the % characters determines how many letters at the start of the word have to match exactly.

Examples:

- *ba%nana* Word must begin with *ba* and have at most one difference between it and *banana*.
- *b%%anana* Word must begin with *b* and have at most two differences between it and *banana*.

Phonic Searching (#)

Phonic searching looks for a word that sounds like the word you are searching for and begins with the same letter. For example, a phonic search for *Smith* will also find *Smithe* and *Smythe*.

To ask the system to search for a word phonically, put a # in front of the word in your search request. Examples: *#smith*, *#johnson*

Phonic searching is somewhat slower than other types of searching and tends to make searches over-inclusive, so it is usually best to use the # symbol selectively.

Stemming (~)

Stemming extends a search to cover grammatical variations on a word. For example, a search for *fish* would also find *fishing*. A search for *applied* would also find *applying*, *applies*, and *apply*.

If you want to add stemming, add a ~ at the end of words that you want stemmed in a search. Example: *apply~*

Numeric Range Searching (~~)

A numeric range search is a search for any numbers that fall within a range. To add a numeric range component to a search request, enter the upper and lower bounds of the search separated by ~~ like this:

- *apple w/5 12~~17*

This request would find any document containing *apple* within 5 words of a number between 12 and 17.

Numeric range searches only work with positive integers. A numeric range search includes the upper and lower bounds (so 12 and 17 would be retrieved in the above example).

For purposes of numeric range searching, decimal points and commas are treated as spaces and minus signs are ignored. For example, -123,456.78 would be interpreted as: 123 456 78 (three numbers).

Ignored Noise Words, Phrases, and Punctuation

Noise Words

Common words such as *if* and *the* are ignored in searches.

Phrases

You do not need to use any special punctuation or commands to search for a phrase. Simply enter the phrase the way it ordinarily appears. You can use a phrase anywhere in a search request. Example:

- apple w/5 fruit salad

If a phrase contains a noise word, this system will skip over the noise word when searching for it. For example, a search for *statue of liberty* would retrieve any document containing the word *statue*, any intervening word, and the word *liberty*.

Punctuation

Punctuation inside of a search word is treated as a space. Thus, *can't* would be treated as a phrase consisting of two words: *can* and *t*. *1843(c)(8)(ii)* would become *1843 c 8 ii* (four words).