

Field Type Analyzer Implementation Guide

Kibo KCCP provides multiple analyzers (also known as "field types") to examine and process field text. These impact indexing and how search results are ranked and displayed. Selecting field and analyzer combinations depends on the fields your customers commonly search for, how each field should be examined, and the importance of each field.

Considerations for Analyzers

Consider the following questions and answers about your search use case to help decide on the best analyzer to use.

Is the search value a title or category name?

- Use `exact_match` to support case-insensitive search with punctuation stripped.
- Add `lenient` if you want to bring in more matches that depend on stemming and synonyms and to support search-as-you-type.
- Add `lenient_phrases` if you want to emphasize phrase matches.

Is the search value a string with a fixed set of possible values? For example, a "Size" field that where values such as "Small" or "Large" are possible. Is it a Brand field that can have specific brand values?

- Use `exact_match` to support a strict matching that is case- and punctuation-insensitive.

Is the search value something that might support a long body of freeform English text? For example, a product description.

- Start with `lenient` to support case-insensitive search with stemming and synonyms.
- Add `lenient_phrases` if you want to emphasize bigram matches.

Example Analyzer Combinations

This section provides examples for how various combinations of analyzers function. Refer to the Field Type Analyzers Description topic for more information on each individual analyzer.

All combinations assume the following search term and possible search results:

Search term: "cat food"

Search result page titles:

1. Cat food
2. Cat foods
3. Food: Cats

When Using: **lenient**

All three pages score the same.

This analyzer performs stemming, which means "food" and "foods" in pages 1 and 2 are given the same consideration for the search term. This analyzer also disregards term order, meaning page 3 is also considered equivalent.

lenient is a good general-purpose analyzer to use for most fields that contain English text.

When Using: **lenient** and **lenient_phrases**

Pages 1 and 2 score highest.

This combination uses the **lenient_phrases** analyzer to match based off term order, then matches based off the **lenient** analyzer. Pages 1 and 2 are matches in both analyzers, while page 3 is a match in just one.

Combining analyzers like this can help score certain results higher while still allowing for a wider breadth of search results.



lenient_phrases is a supporting analyzer. If you use it, you must also use **lenient**.

When Using: **exact_match**

Page 1 scores the highest. Pages 2 and 3 score the same.

This analyzer does not perform stemming, which means "food" and "foods" in pages 1 and 2 are considered distinct terms. This analyzer also considers term order, meaning page 1 is a more exact match than page 3.

exact_match is best used when you want to show matches that are exactly the same as what your customer types as a search term. This analyzer is not recommended on its own if you want search results to have more leeway.

When Using: **lenient** and **exact_match**

Page 1 scores the highest. Pages 2 and 3 score the same.

This combination uses the **exact_match** analyzer to look for exact matches, then matches off the **lenient** analyzer. Page 1 is a match for both analyzers, while pages 2 and 3 are matches in just one.

When Using: **lenient**, **exact_match**, and **lenient_phrases**

Page 1 scores the highest. Page 2 scores the second highest. Page 3 scores the lowest.

Like previous combination examples, results are scored by how many analyzers match. In this case, page 1 matches all three analyzers because it is an exact term match with exact term ordering. Page 2 scores behind this because the need for stemming prevents a match with `exact_match`. Page 3 scores lowest because of the need for stemming and because the term order prevents a match with `lenient_phrases`.