

Configure BM25 relevance scoring

In this article, learn how to configure the [BM25 relevance scoring algorithm](#) used by Azure AI Search for full text search queries. It also explains how to enable BM25 on older search services.

BM25 applies to:

- Queries that use the [search](#) parameter for full text search, on text fields having a [searchable](#) attribution.
- Scoring is scoped to [searchFields](#), or to all [searchable](#) fields if [searchFields](#) is null.

The search engine uses BM25 to calculate a **@searchScore** for each match in a given query. Matching documents are ranked by their search score, with the top results returned in the query response. It's possible to get some [score variation](#) in results, even from the same query executing over the same search index, but usually these variations are small and don't change the overall ranking of results.

BM25 has defaults for weighting term frequency and document length. You can customize these properties if the defaults aren't suited to your content. Configuration changes are scoped to individual indexes, which means you can adjust relevance scoring based on the characteristics of each index.

Default scoring algorithm

Depending on the age of your search service, Azure AI Search supports two [scoring algorithms](#) for a full text search query:

- Okapi BM25 algorithm (after July 15, 2020)
- Classic similarity algorithm (before July 15, 2020)

BM25 ranking is the default because it tends to produce search rankings that align better with user expectations. It includes [parameters](#) for tuning results based on factors such as document size. For search services created after July 2020, BM25 is the only scoring algorithm. If you try to set "similarity" to ClassicSimilarity on a new service, an HTTP 400 error is returned because that algorithm isn't supported by the service.

For older services, classic similarity remains the default algorithm. Older services can [upgrade to BM25](#) on a per-index basis. When switching from classic to BM25, you can expect to see some differences how search results are ordered.

Set BM25 parameters

BM25 ranking provides two parameters for tuning the relevance score calculation.

1. Use a [Create or Update Index](#) request to set BM25 parameters:

```
PUT [service-name].search.windows.net/indexes/[index-name]?api-version=2024-07-01&allowIndexDowntime=true
{
  "similarity": {
```

```
    "@odata.type": "#Microsoft.Azure.Search.BM25Similarity",
    "b" : 0.75,
    "k1" : 1.2
  }
}
```

2. If the index is live, append the `allowIndexDowntime=true` URI parameter on the request, shown on the previous example.

Because Azure AI Search doesn't allow updates to a live index, you need to take the index offline so that the parameters can be added. Indexing and query requests fail while the index is offline. The duration of the outage is the amount of time it takes to update the index, usually no more than several seconds. When the update is complete, the index comes back automatically.

3. Set `"b"` and `"k1"` to custom values, and then send the request.

Property	Type	Description
k1	number	Controls the scaling function between the term frequency of each matching terms to the final relevance score of a document-query pair. Values are usually 0.0 to 3.0, with 1.2 as the default.
		A value of 0.0 represents a "binary model", where the contribution of a single matching term is the same for all matching documents, regardless of how many times that term appears in the text. Larger k1 values allow the score to continue to increase as more instances of the same term is found in the document.
		Using a larger k1 value is important in cases where multiple terms are included in a search query. In those cases, you might want to favor documents matching more query terms, over documents that only match a single term, multiple times. For example, when querying for the terms "Apollo Spaceflight", you might want to lower the score of an article about Greek Mythology that contains the term "Apollo" a few dozen times, without mentions of "Spaceflight", relative to another article that explicitly mentions both "Apollo" and "Spaceflight" a handful of times only.
b	number	Controls how the length of a document affects the relevance score. Values are between 0 and 1, with 0.75 as the default.
		A value of 0.0 means the length of the document doesn't influence the score. A value of 1.0 means the effect of term frequency on relevance score is normalized by the document's length.
		Normalizing the term frequency by the document's length is useful in cases where you want to penalize longer documents. In some cases, longer documents (such as a complete novel), are more likely to contain many irrelevant terms, compared to shorter documents.

Enable BM25 scoring on older services

If you're running a search service that was created from March 2014 through July 15, 2020, you can enable BM25 by setting a "similarity" property on new indexes. The property is only exposed on new indexes, so if you want BM25 on an existing index, you must drop and [rebuild the index](#) with a "similarity" property set to `Microsoft.Azure.Search.BM25Similarity`.

Once an index exists with a "similarity" property, you can switch between `BM25Similarity` or `ClassicSimilarity`.

The following links describe the Similarity property in the Azure SDKs.

Client library	Similarity property
.NET	SearchIndex.Similarity
Java	SearchIndex.setSimilarity
JavaScript	SearchIndex.Similarity
Python	similarity property on SearchIndex

REST example

You can also use the [REST API](#). The following example creates a new index with the "similarity" property set to BM25:

```
PUT [service-name].search.windows.net/indexes/[index name]?api-version=2024-07-01
{
  "name": "indexName",
  "fields": [
    {
      "name": "id",
      "type": "Edm.String",
      "key": true
    },
    {
      "name": "name",
      "type": "Edm.String",
      "searchable": true,
      "analyzer": "en.lucene"
    },
    ...
  ],
  "similarity": {
    "@odata.type": "#Microsoft.Azure.Search.BM25Similarity"
  }
}
```

See also

- [Relevance and scoring in Azure AI Search](#)
- [REST API Reference](#)
- [Add scoring profiles to your index](#)
- [Create Index API](#)
- [Azure AI Search .NET SDK](#)