RediSearch 1.2 release notes

Requirements

RediSearch v1.2.0 requires:

- Minimum Redis compatibility version (database): 4.0.0
- Minimum Redis Enterprise Software version (cluster): 5.0.0

RediSearch 1.2.0 (June 2018)

This version is the last version that I (@dvirsky) will be releasing, and it includes a lot of new cool features, and actually NO bug-fixes over 1.1.0! So long and thanks for all the fish!

New features

Aggregation filters

As an addition to the aggregation API in FT.AGGREGATE, it is possible to do post-index filtering of the pipeline, using the FILTER keyword. e.g.:

```
FT.AGGREGATE idx "*"
  GROUPBY 1 @foo
  REDUCE count 0 AS num
  FILTER "@num < 100"
```

See http://redisearch.io/Aggregations/ for more details.

Query attributes

It is now possible to apply specific query modifying attributes to specific clauses of the query (see #212).

The syntax is (foo bar) => { $attribute: value; $attribute:value; ...}, e.g:

```
(foo bar) => { $weight: 2.0; $slop: 1 }
~(bar baz) => { $weight: 0.5; }
```

The supported attributes are:

1. $weight: determines the weight of the sub-query or token in the overall ranking on the result.
2. $slop: determines the maximum allowed "slop" (space between terms) in the query clause.
3. $inorder: whether or not the terms in a query clause must appear in the same order as in the query.

Fuzzy matching

Wrapping a search term with % will cause the index to expand the query to terms that are within an Edit Distance of 1 from the original term. For example, querying for %redis% will expand it to query for redis, jedis, credis, predis, etc (provided the terms appear in documents in the index).
Notice that each query term needs to be wrapped independently, and that we limit the maximum amount of expansions to 200 per term, as this hurts performance significantly.

**Conditional updates**

It is now possible to update documents (FT.ADD ... REPLACE [PARTIAL]) only if a certain condition is met regarding the document's state before the updates. So for example, if our document has a timestamp field, and we would like to update its title only if the timestamp is below a certain value, we can do the following:

```plaintext
FT.ADD myIndex myDoc 1.0
REPLACE PARTIAL
IF "@timestamp < 12313134523"
FIELDS
title "new title"
```

**Backslash escaping**

Following several user requests, it is now possible to escape separator characters in documents (it is already done in the query itself), and avoid tokenization when needed.

For example, indexing the text `hello-world` `hello world` will create the tokenization `"hello-world", "hello", "world"`. Notice that in most languages and in redis-cli, you will need to escape the backslash itself, so instead of `hello-world` you will need to send the string `hello\-world`.

The same goes to the query string: If the document contains the token `hello-world`, it can be found by running in redis-cli: `FT.SEARCH idx "hello-world", which will cause the query not to be separated.

**Synonyms support**

It is now possible to provide the index with synonym groups (e.g. boy, child, kid), and have it automatically index synonymous terms so that searching for one will return documents containing another.


---

**Updated:** December 14, 2021