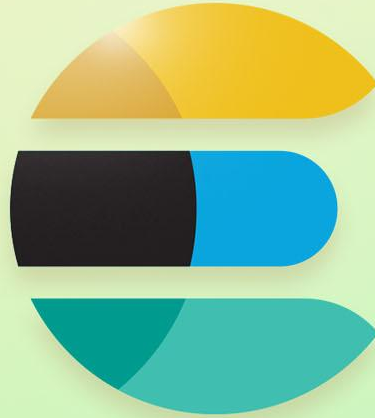# boosting query

Introduction

- The `bool` query enables us to increase relevance scores with `should`

- What if we want to *decrease* relevance scores for some documents?

  - This can be done with the `boosting` query

"I like juice, but not if it contains apples"

```
GET /products/_search
{
  "query": {
    "boosting": {
      "positive": {
        "match": {
          "name": "juice" 🔵
        }
      },
      "negative": {
        "match": {
          "name": "apple" 🔴
        }
      },
      "negative_boost": 0.5
    }
  }
}
```

```
{
    "name": "Orange Juice",
    "_score": 5.057956
}
```

5.057956
N/A

= 5.057956

```
{
    "name": "Juice - Mango",
    "_score": 4.9778986
}
```

4.9778986
N/A

= 4.9778986

```
{
    "name": "Cranberry Juice",
    "_score": 4.9778986
}
```

4.9778986
N/A

= 4.9778986

```
{
    "name": "Apple Juice",
    "_score": 2.528978
}
```

5.057956
× 0.5

= 2.528978

```
{
    "name": "Juice - Apple Cider",
    "_score": 2.172309
}
```

4.344618
× 0.5

= 2.172309

"I don't like apples in general" #nofilter

```
GET /products/_search
{
  "query": {
    "boosting": {
      "positive": {
        "match_all": { }  🔵
      },
      "negative": {
        "match": {
          "name": "apple"  🔴
        }
      },
      "negative_boost": 0.5
    }
  }
}
```

```
{
  "name": "Nestea - Iced Tea",
  "_score": 1
}
```

| 1 |
| N/A |

= 1

```
{
  "name": "Garlic",
  "_score": 1
}
```

| 1 |
| N/A |

= 1

```
{
  "name": "Toaster",
  "_score": 1
}
```

| 1 |
| N/A |

= 1

```
{
  "name": "Apple Juice",
  "_score": 0.5
}
```

| 1 |
| × 0.5 |

= 0.5

```
{
  "name": "Juice - Apple Cider",
  "_score": 0.5
}
```

| 1 |
| × 0.5 |

= 0.5

CodingExplained
CODE. SLEEP. REPEAT.

Query examples

"I like pasta"

```
GET /recipes/_search
{
  "query": {
    "bool": {
      "must": [
        { "match_all": {} }
      ],
      "should": [
        {
          "term": {
            "ingredients.name.keyword": "Pasta" (1)
          }
        }
      ]
    }
  }
}
```

"I don't like bacon"

```
GET /recipes/_search
{
  "query": {
    "boosting": {
      "positive": {
        "match_all": {}
      },
      "negative": {
        "term": {
          "ingredients.name.keyword": "Bacon" (2)
        }
      },
      "negative_boost": 0.5
    }
  }
}
```

(1) Increase relevance scores of recipes containing pasta

(2) Reduce relevance scores of recipes containing bacon

CodingExplained
CODE. SLEEP. REPEAT.

"I like pasta, but not with bacon"

```
GET /recipes/_search
{
  "query": {
    "boosting": {
      "positive": {
        "term": {
          "ingredients.name.keyword": "Pasta"
        }
      },
      "negative": {
        "term": {
          "ingredients.name.keyword": "Bacon"
        }
      },
      "negative_boost": 0.5
    }
  }
}
```

"I like pasta, but not bacon"

```
GET /recipes/_search
{
  "query": {
    "boosting": {
      "positive": {
        "bool": {
          "must": [
            { "match_all": {} }
          ],
          "should": [
            {
              "term": {
                "ingredients.name.keyword": "Pasta"
              }
            }
          ]
        }
      },
      "negative": {
        "term": {
          "ingredients.name.keyword": "Bacon"
        }
      },
      "negative_boost": 0.5
    }
  }
}
```

CodingExplained
CODE. SLEEP. REPEAT.

Lecture summary

- The `bool` query can increase relevance scores with `should`

- The `boosting` query can *decrease* relevance scores with `negative`

  - Documents must match the `positive` query clause

  - Documents that match the `negative` query clause have their relevance scores decreased

  - Use `match_all` query for `positive` if you don't want to filter documents

  - Can be used with any query (including compound queries, such as `bool`)

CodingExplained
CODE. SLEEP. REPEAT.