

14 - Sorting results, pagination, highlighting

Ve výchozím stavu jsou výsledky vyhledávání řazeny podle jejich skóre. Skóre (`_score`) je desetinné číslo, které nám říká, jak nalezený dokument odpovídá vyhledávacímu dotazu. Skóre záleží na "term frequency" a "inverse document frequency".

Term frequency

Frekvence vyhledávaného termu v dokumentu ovlivňuje výsledné skóre:

```
POST sorting_1/_bulk
{"index":{}}
{"title": "Black Black Black"}
{"index":{}}
{"title": "Black"}
{"index":{}}
{"title": "Black"}
```

Když vyhledáme `Black`:

```
GET sorting_1/_search
{
  "query": {
    "match": {
      "title": "Black"
    }
  }
}
```

Nejvyšší skóre bude mít dokument, který obsahuje slovo `Black` třikrát:

```
{
  "_score" : 0.18952844,
  "_source" : {
    "title" : "Black Black Black"
  }
},
{
  "_score" : 0.14874382,
  "_source" : {
    "title" : "Black"
  }
},
```

```
{
  "_score" : 0.14874382,
  "_source" : {
    "title" : "Black"
  }
}
```

Inverse document frequency

Dále je důležité, jak unikátní je term v rámci indexu. Méně unikátní term znamená menší skóre. Když vyhledáme výraz `Black White` v následujících dokumentech:

```
POST sorting_2/_bulk
{"index":{}}
{"title": "Black"}
{"index":{}}
{"title": "Black"}
{"index":{}}
{"title": "White"}

GET sorting_2/_search
{
  "query": {
    "match": {
      "title": "Black White"
    }
  }
}
```

Nejvyšší skóre bude mít dokument obsahující unikátní term `White`:

```
{
  "_score" : 0.9808291,
  "_source" : {
    "title" : "White"
  }
},
{
  "_score" : 0.4700036,
  "_source" : {
    "title" : "Black"
  }
},
{
  "_score" : 0.4700036,
  "_source" : {
```

```
    "title" : "Black"
  }
}
```

Field length normalization

Čím vyšší procento termů v dokumentu odpovídá vyhledávacímu dotazu, tím vyšší bude skóre:

```
POST sorting_3/_bulk
{"index":{}}
{"title": "Happy black dog"}
{"index":{}}
{"title": "Black dog"}

GET sorting_3/_search
{
  "query": {
    "match": {
      "title": "Black"
    }
  }
}

// Result:
{
  "_score" : 0.19856803,
  "_source" : {
    "title" : "Black dog"
  }
},
{
  "_score" : 0.16853255,
  "_source" : {
    "title" : "Happy black dog"
  }
}
```

Coordination

Více termů nalezených v dokumentu znamená opět vyšší skóre:

```
POST sorting_4/_bulk
{"index":{}}
{"title": "Black black"}
```

```
{ "index": {} }
{ "title": "Black dog" }
{ "index": {} }
{ "title": "Dog dog" }

GET sorting_4/_search
{
  "query": {
    "match": {
      "title": "Black dog"
    }
  }
}

// Results:
{
  "_score" : 0.9400072,
  "_source" : {
    "title" : "Black dog"
  }
},
{
  "_score" : 0.646255,
  "_source" : {
    "title" : "Black black"
  }
},
{
  "_score" : 0.646255,
  "_source" : {
    "title" : "Dog dog"
  }
}
```

Custom sorting

Pokud chcete řadit výsledky podle jiného pole, než podle výsledného skóre, je to možné určit pomocí parametru `sort`:

```
POST sorting_5/_bulk
{ "index": {} }
{ "title": "Black dog", "comments": 100 }
{ "index": {} }
{ "title": "Black cat", "comments": 50 }
{ "index": {} }
{ "title": "Black dog", "comments": 50 }
```

```
GET sorting_5/_search
{
  "sort": [
    {
      "comments": "desc"
    },
    {
      "title.keyword": "asc"
    }
  ]
}

// Results:
{
  "_source" : {
    "title" : "Black dog",
    "comments" : 100
  },
  "sort" : [
    100,
    "Black dog"
  ]
},
{
  "_source" : {
    "title" : "Black cat",
    "comments" : 50
  },
  "sort" : [
    50,
    "Black cat"
  ]
},
{
  "_source" : {
    "title" : "Black dog",
    "comments" : 50
  },
  "sort" : [
    50,
    "Black dog"
  ]
}
```

Boostování konkrétních polí v `multi_match` query

V případě `multi_match` query je možné určit pomocí znaku `^` které pole mají vyšší (nebo naopak nižší) prioritu než ostatní:

```
POST sorting_6/_doc
{
  "title": "mobile phone apple iphone X 64GB",
  "description": "bla bla bla"
}

POST sorting_6/_doc
{
  "title": "usb cable",
  "description": "compatible with apple iphone"
}

GET sorting_6/_search
{
  "query": {
    "multi_match": {
      "query": "apple iphone",
      "fields": [
        "title",
        "description^0.5"
      ],
      "type": "most_fields"
    }
  }
}
```

Kombinování skóre a popularity

V některých případech chceme mírně upravit pořadí výsledků, které jsou jinak řazeny dle skóre. V případě produktů by to mohla být prodejnost, nebo například marže. Při vyhledávání na webu obdobným způsobem do pořadí výsledků vstupuje page rank.

Toho lze dosáhnout pomocí `rank_score` query (v kombinaci s datovým typem `rank_features`):

```
PUT my_sorting
{
  "mappings": {
    "properties": {
      "popularity": {
        "type": "rank_features"
      }
    }
  }
}
```

```
POST my_sorting/_doc
{
  "title": "Samsung TV AL32B2019VQX",
  "ranking": {
    "user_popularity": 3
  }
}

POST my_sorting/_doc
{
  "title": "Samsung TV BL40B2020V",
  "ranking": {
    "user_popularity": 9
  }
}

GET _search
{
  "query": {
    "bool": {
      "must": {
        "match": {
          "title": "samsung"
        }
      },
      "should": {
        "rank_feature": {
          "field": "ranking.user_popularity"
        }
      }
    }
  }
}
```

Případně pro vyšší flexibilitu lze použít `script_score`:

```
GET _search
{
  "query": {
    "script_score": {
      "query": {
        ...
      },
      "script": {
        "source": "_score * doc['popularity'].value"
      }
    }
  }
}
```

Stránkování

Stránkovat výsledky lze pomocí parametrů `from` a `size`. výchozí hodnota pro `size` je `10`.

```
GET _search
{
  "from": 10,
  "size": 20
}
```

Limit při stránkování je nastaven na 10 000 výsledků. Změnit jej lze v nastavení Elasticsearch pod `index.max_result_window`. Pokud ale potřebujete projít opravdu velké množství dat, doporučuji spíše použít `search_after` ([docs](#)).

Zvýrazňování

Jak můžete vidět v Discover v Kibaně, nalezené výsledky mohou používat zvýrazňování.

To lze uvést na nejvyšší úrovni query pod klíčem `highlight`. Je možné určit, čím přesně bude zvýraznění probíhat (typicky HTML tagy):

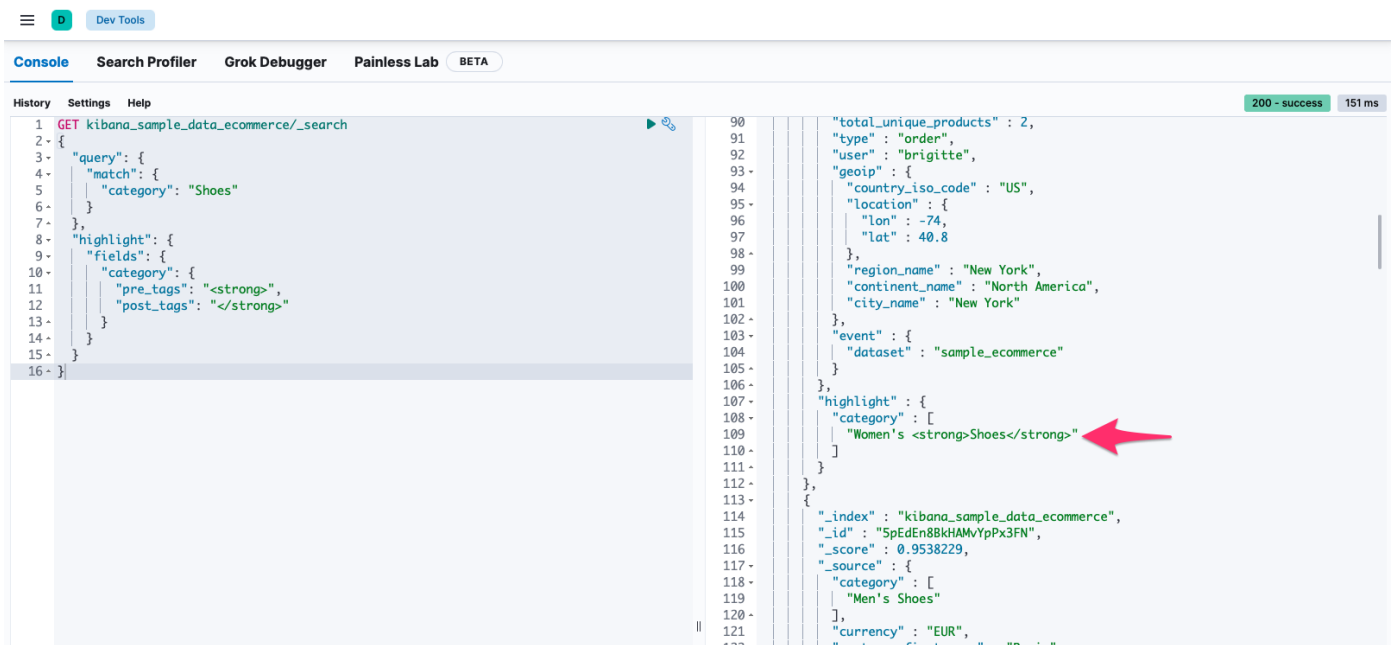
```
GET kibana_sample_data_ecommerce/_search
{
  "query": {
    "match": {
      "category": "Shoes"
    }
  },
  "highlight": {
    "fields": {
```



```

    "category": {
      "pre_tags": "<strong>",
      "post_tags": "</strong>"
    }
  }
}
}

```



Úkol: sorting, pagination, and highlighting

1. Uložte následující dokumenty do Elasticsearch:

```
POST book/_doc
```

```
{
  "title": "The Forever Dog",
  "publish": "2020-01-20"
}
```

```
POST book/_doc
```

```
{
  "title": "The Book Your Dog Wishes",
  "publish": "2018-01-01"
}
```

```
POST book/_doc
```

```
{
  "title": "The Complete Dog Breed Book",
  "publish": "2018-01-01"
}
```

2. Vyhledejte `Dog`, přičemž:

1. Seřadte výsledky podle data `publish`, od nejstarších k nejnovějším
2. Pokud mají dva dokumenty shodné datum, seřadte je abecedně
3. Zvýrazněte shodu pomocí HTML tagu ``