

9 텍스트 검색

엔지니어링 12기 박다정

목차

- 텍스트 검색이 중요한 이유
- 텍스트 검색 기본
- MongoDB 텍스트 검색 인덱스 정의
- MongoDB find()로 텍스트 검색 사용하기
- 집계와 함께 MongoDB 텍스트 검색 사용하기
- 다른 언어로 텍스트 검색 사용하기

9.1 텍스트 검색

1. 텍스트 검색 vs. 패턴 일치

- Case-insensitive

Java => java, jAVA

- Stemming

검색어, 검색어가 파생되는 줄기 또는 뿌리 단어로 변환

script => scripting, scripts, scripted (O) JavaScript (X)

9.1 텍스트 검색

2. 텍스트 검색 vs. 웹 페이지 검색

- Page Rank

링크된 페이지의 중요도를 토대로 페이지의 중요도/가중치 평가

9.1 텍스트 검색

3. MongoDB 텍스트 검색 vs. 전용 텍스트 검색 엔진

텍스트 검색 엔진 (맞춤법 교정, 제안, 관련성 측정)

+

전용 텍스트 검색 엔진

(facets, 사용자 정의 동의어 라이브러리, 형태소 분석, 불용어)

The screenshot shows a search results page for '올리브데올리브' (Olive de Olive) on the Shopee platform. At the top, there's a search bar with the query '올리브데올리브'. Below the search bar, the navigation bar includes categories like 전체, 패션/뷰티, 가전/컴퓨터, 가구/생활/건강, 식품/유아동, 여행/레저/자동차, and a 'My' dropdown. A red box highlights the '관련검색어' (Related Searches) section, which lists terms such as 온라인, 올리브데올리브 블라우스, 올리브데올리브 원피스, 올리브데올리브 코트, 나이스크립, 올리브 데 올리브 니트, 아울렛, 올리브데올리브 티셔츠, 올리브데올리브 페딩, and 더보기.

In the center, a blue banner indicates '올리브데올리브 검색결과입니다. (카테고리 33 / 상품수 93,468)'. To the right, there's a '결과 내 재검색' (Search again within results) button. Below the banner, a red box highlights the '카테고리' (Category) facet filter, which lists items like 자켓/점퍼, 원피스, 블라우스/셔츠, 모피카디/코트, 스커트, 니트/스웨터, 바지, 티셔츠, 가디건, 조끼, 마플리스카프/슬, and 카테고리 더보기.

At the bottom of the facet filter, there are sections for '브랜드' (Brand), '가격' (Price), and '컬러' (Color). The '가격' section shows filters for 4~6만원, 6~9만원, 9~12만원, 12~16만원, 16~20만원, 20~24만원, 800 원 ~ 3,414,300 원, and a '검색' (Search) button. The '컬러' section shows four color swatches: red, green, blue, and white.

At the very bottom, there are several buttons: '전체' (All) with 93,468 items, '가격비교' (Price Comparison) with 895 items, '일반상품' (General Products) with 92,293 items, '해외구매' (Overseas Purchase) with 280 items, a '40개씩' (40 items each) dropdown, and a grid icon. The footer contains links for '쇼핑하우 향강순', '낮은가격순', '높은가격순', '상품평순', '만족도순', and '최신순'. It also includes filters for '쇼핑몰 선택', '할인/조건', and '무료배송'.

9.2 텍스트 검색 인덱스의 정의

무엇을 제공?

- 형태소 분석을 통한 자동 실시간 인덱싱
- 필드 이름을 통해 선택적으로 지정한 가능한 가중치
- 다국어 지원
- 불용어 삭제
- 정확한 구문 또는 단어 일치
- 주어진 구 또는 단어로 결과를 제외할 수 있는 기능

어떻게 제공?

1. 텍스트 검색에 필요한 인덱스를 정의한다
2. 집계 프레임워크와 기본 쿼리 모두에서 텍스트 검색을 사용한다.

9.2 텍스트 검색 인덱스의 정의

- 필드가 인덱싱된 후에 1또는 -1을 지정하는 대신 텍스트를 사용한다.
- 텍스트 인덱스의 일부가 될 필드를 지정할 수 있으며, 모든 필드는 단일 필드인 것처럼 함께 검색된다.
- 컬렉션당 하나의 텍스트 검색 인덱스만 가질 수 있지만, 원하는 만큼 필드를 인덱싱할 수 있다.

가중치: 필드가 얼마나 중요한지 지정하여 검색 결과를 점수화하기 위함

```
db.books.createIndex(  
    {title: 'text',  
     shortDescription: 'text',  
     longDescription: 'text',  
     authors: 'text',  
     categories: 'text'},  
  
    {weights:  
        {title: 10,  
         shortDescription: 1,  
         longDescription: 1,  
         authors: 1,  
         categories: 5}  
    }  
);
```

텍스트 인덱싱할 필드를 지정한다

선택적으로 각 필드의 가중치를 지정한다

9.2 텍스트 검색 인덱스의 정의

1. 텍스트 인덱스 크기

- 인덱스 항목수를 줄이기 위해 불용어는 무시된다 (the, an, a, and)
- stat(): 컬렉션의 크기와 인덱스의 크기를 보여줌

```
> db.books.stats()
{
  "ns" : "catalog.books",
  "count" : 431,
  "size" : 772368,
  "avgObjSize" : 1792,
  "storageSize" : 2793472,
  "numExtents" : 5,
  "nindexes" : 2,
  "lastExtentSize" : 2097152,
  "paddingFactor" : 1,
  "systemFlags" : 0,
  "userFlags" : 1,
  "totalIndexSize" : 858480,
  "indexSizes" : {
    "_id_" : 24528,
    "title_text_shortDescription_text_longDescription_text_authors_text
    _categories_text" : 833952
  },
  "ok" : 1
}
```

books 컬렉션의 크기

텍스트 검색 인덱스의 이름과 크기

- 텍스트 검색 인덱스 (833,952) > 컬렉션 (772,368)
: 인덱스가 생성되는 대부분의 텍스트를 복제할 뿐만 아니라 각 단어의 원본 도큐먼트에 대한 포인터를 추가해야 한다
- 인덱스 이름의 길이
: 최대 길이 123바이트
-> 인덱스에 사용자 정의 이름을 할당하자

9.2 텍스트 검색 인덱스의 정의

2. 인덱스 이름 지정 및 컬렉션의 모든 텍스트 필드 인덱싱

- 가중치 default = 1
- 이미 인덱스가 존재하는 경우에는 dropIndex()로 삭제 후 생성

```
db.books.createIndex(  
    {title: 'text',  
     shortDescription: 'text',  
     longDescription: 'text',  
     authors: 'text',  
     categories: 'text'},  
    {weights:  
        {title: 10,  
         categories: 5},  
  
        name : 'books_text_index'  
    }  
);
```

필드에 1 이외의
가중치를 지정

사용자 정의
인덱스 이름

9.2 텍스트 검색 인덱스의 정의

2. 인덱스 이름 지정 및 컬렉션의 모든 텍스트 필드 인덱싱

와일드카드 필드 이름

: 문자열이 들어 있는 필드를 인덱싱

'\$**': 'text'

-> name : '\$**_text'

```
db.books.createIndex(  
  {'$**': 'text'},  
  {weights:  
    {title: 10,  
     categories: 5},  
  });
```

모든 필드를 문자열로
인덱싱한다

9.3 기본 텍스트 검색

```
db.books.find({$text: {$search: 'actions'}}, {title:1})
```

```
{ "_id" : 256, "title" : "Machine Learning in Action" }      stemming  
{ "_id" : 146, "title" : "Distributed Agile in Action" }  
{ "_id" : 233, "title" : "PostGIS in Action" }  
{ "_id" : 17, "title" : "MongoDB in Action" }  
...
```

- \$text: 쿼리를 텍스트 검색으로 정의
 - \$search: 검색에 사용할 문자열 정의
- 모든 텍스트 필드를 스캔하는 대신 인덱스를 사용하여 찾으므로 빠르다

9.3 기본 텍스트 검색

```
db.books.find({$text: {$search: 'MongoDB in Action'}}, {title:1})
```

```
{ "_id" : 256, "title" : "Machine Learning in Action" }  
{ "_id" : 146, "title" : "Distributed Agile in Action" }  
{ "_id" : 233, "title" : "PostGIS in Action" }  
{ "_id" : 17, "title" : "MongoDB in Action" }  
...
```

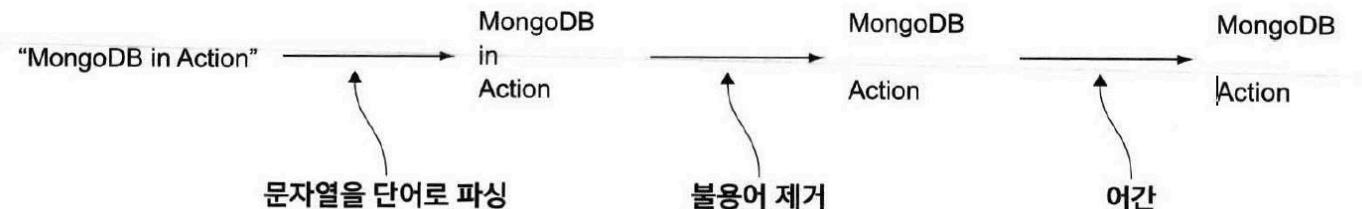


그림 9.7 텍스트 검색 문자열 처리

'MongoDB' 검색한 다음 다시 'action' 검색

9.3 기본 텍스트 검색

1. 더 복잡한 검색

1. or 단어 매칭 대신 and 단어 일치를 지정한다
2. 정확한 구문 매칭을 수행한다
3. 특정 단어가 있는 도큐먼트를 제외한다
4. 특정 문구가 포함되어 있는 도큐먼트를 제외한다

9.3 기본 텍스트 검색

1. 더 복잡한 검색

1. or 단어 매칭 대신 and 단어 일치를 지정한다

```
db.books.
```

```
  find({$text: {$search: ' "mongodb" in action'}}) ←
```

큰따옴표 안의 mongodb는 해당 단어가
결과 도큐먼트에 있어야 함을 의미한다

```
{ "title" : "MongoDB in Action"}  
{ "title" : "MongoDB in Action, Second Edition" }
```

9.3 기본 텍스트 검색

1. 더 복잡한 검색

2. 정확한 구문 매칭을 수행한다.

```
> db.books.  
...   find({$text: {$search: 'mongodb "second edition" '}},  
...   {_id:0, title:1})  
{ "title" : "MongoDB in Action, Second Edition" }
```

['mongodb'라는 단어뿐만 아니라 'second edition'이라는 구도 필요하다.]

```
> db.books.  
...   find({$text: {$search: ' books '}}). ← 단어 'books'의  
...   count()                           축약 버전 -414
```

414

>

```
> db.books.  
...   find({$text: {$search: ' "books" '}}). ← 단어 'books'의  
...   count()                           정확한 버전 -21
```

21

9.3 기본 텍스트 검색

1. 더 복잡한 검색

3. 특정 단어가 있는 도큐먼트를 제외한다

```
> db.books.  
...     find({$text: {$search: ' mongodb -second '}}, ← 'second'라는 단어를 가진  
...     {_id:0, title:1 })                           도큐먼트를 제외한다  
{ "title" : "MongoDB in Action" }
```

4. 특정 문구가 포함된 도큐먼트를 제외한다

```
> db.books.  
...     find({$text: {$search: ' mongodb -"second edition" '}}, ← 'second edition'  
...     {_id:0, title:1 })                           이라는 구를 가진  
{ "title" : "MongoDB in Action" }                   도큐먼트를 제외한다
```

9.3 기본 텍스트 검색

1. 더 복잡한 검색

더욱 복잡한 검색 명세

```
> db.books.  
...     find({$text: {$search: ' mongodb '}, status: 'MEAP' }, ←  
...     {_id:0, title:1, status:1})  
{ "title" : "MongoDB in Action, Second Edition",  
"status" : "MEAP"}
```

상태(status)가
MEAP이어야 한다

9.3 기본 텍스트 검색

2. 텍스트 검색 스코어

단어가 도큐먼트에 표시된 횟수를 기반으로 도큐먼트의 관련성을 평가하는 숫자 제공
필드에 할당된 가중치도 사용

```
> db.books.  
...   find({$text: {$search: 'mongodb in action'}},      ← 'MongoDB in action'을 검색한다  
...   {_id:0, title:1, score: { $meta: "textScore" }}).    ← 결과에 텍스트 검색  
...   limit(4);  
{ "title" : "Machine Learning in Action", "score" : 16.83933933933934 }  
{ "title" : "Distributed Agile in Action", "score" : 19.371088861076345 }  
{ "title" : "PostGIS in Action", "score" : 17.67825896762905 }      ← 첫 번째 검색  
{ "title" : "MongoDB in Action", "score" : 49.48653394500073 }      ← 문자열에 대한  
                                                               텍스트 검색 스코어  
>  
>  
> db.books.  
...   find({$text: {$search: 'the mongodb and actions in it'}},      ←  
...   {_id:0, title:1, score: { $meta: "textScore" }}).  
...   limit(4);  
{ "title" : "Machine Learning in Action", "score" : 16.83933933933934 }  
{ "title" : "Distributed Agile in Action", "score" : 19.371088861076345 }  
⇒ { "title" : "PostGIS in Action", "score" : 17.67825896762905 }  
{ "title" : "MongoDB in Action", "score" : 49.48653394500073 }
```

두 번째 문자열 텍스트 점수는 -
첫 번째 점수 집합과 동일하다

추가적인 불용어와 복수 단어 'actions'가
있는 두 번째 텍스트 문자열

불용어 무시, stemming 확인

9.3 기본 텍스트 검색

3. 텍스트 검색 스코어에 의한 결과 정렬

단어가 도큐먼트에 표시된 횟수를 기반으로 도큐먼트의 관련성을 평가하는 숫자 제공
필드에 할당된 가중치도 사용

```
db.books.  
  find({$text: {$search: 'mongodb in action'}},  
        {title:1, score: { $meta: "textScore" }}). ← 텍스트 스코어에  
  sort({ score: { $meta: "textScore" } })      ← 대한 프로젝션 텍스트 스코어로 정렬
```

```
{ "_id" : 17, "title" : "MongoDB in Action", "score" : 49.48653394500073 }  
{ "_id" : 186, "title" : "Hadoop in Action", "score" : 24.99910329985653 }  
{ "_id" : 560, "title" : "HTML5 in Action", "score" : 23.02156177156177 }
```

9.4 집계 프레임워크 텍스트 검색

6장에서 배운 집계 프레임워크 사용

```
db.books.aggregate(  
  [  
    { $match: { $text: { $search: 'mongodb in action' } } }, ← mongodb 또는  
    { $sort: { score: { $meta: 'textScore' } } }, ← action이라는 단어로 된  
    { $project: { title: 1, score: { $meta: 'textScore' } } } ← 도큐먼트를 검색한다  
  ] ← 텍스트 스코어로  
) ← 정렬 ← 텍스트 스코어에  
          대한 프로젝션
```

```
db.books.aggregate(  
  [  
    { $match: { $text: { $search: 'mongodb in action' } } }, ← 스코어  
    { $project: { title: 1, score: { $meta: 'textScore' } } }, ← 내림차순  
    { $sort: { score: -1 } } -1: descending ← 정렬  
  ] ← 1:ascending  
)
```

\$text 검색의 제한 사항

- \$text 함수 검색을 사용하는 \$match는 파이프라인의 첫번째 연산이어야 하고, \$meta: 'textScore'에 대한 다른 참고 앞에 와야 한다
- \$text 함수는 파이프라인에 한 번만 나타날 수 있다
- \$text 함수는 \$or 또는 \$not과 함께 사용할 수 없다

9.5 텍스트 검색 언어

- 인덱스에서:
특정 컬렉션에 대한 기본 언어를 지정할 수 있다
- 도큐먼트를 삽입할 때:
MongoDB에 도큐먼트 내의 특정 도큐먼트 또는 필드가 인덱스 지정 기본값 이외의 다른 언어임을 알리기 위해 이 기본값을 오버라이드 할 수 있다
- `find()` 또는 `aggregate()` 함수에서 텍스트 검색을 수행할 때:
검색이 사용하는 언어를 MongoDB에 알릴 수 있다

9.5 텍스트 검색 언어

- 인덱스에서:
특정 컬렉션에 대한 기본 언어를 지정할 수 있다

```
db.books.dropIndex('books_text_index'); ← books 컬렉션에 있던 기존의 텍스트 인덱스를 삭제한다  
  
db.books.createIndex(  
  {'$**': 'text',  
  
  {weights:  
    {title: 10,  
     categories: 5},  
  
  name : 'books_text_index',  
  
  default_language: 'french'  
}  
);
```

```
> db.books.find({$text: {$search: 'in'}}).count()  
0
```

영어에서는 'in'이 불용어

```
> db.books.find({$text: {$search: 'in'}}).count()  
334
```

프랑스어에서는 'in'이 불용어가 아님

9.5 텍스트 검색 언어

- 도큐먼트를 삽입할 때:

MongoDB에 도큐먼트 내의 특정 도큐먼트 또는 필드가 인덱스 지정 기본값 이외의 다른 언어임을 알리기 위해 이 기본값을 오버라이드 할 수 있다

```
db.books.insert({  
    _id: 999,  
    title: 'Le Petit Prince',  
    pageCount: 85,  
    publishedDate: ISODate('1943-01-01T01:00:00Z'),  
    shortDescription: "Le Petit Prince est une oeuvre de langue française,  
    la plus connue d'Antoine de Saint-Exupéry. Publié en 1943 à New York  
    simultanément en anglais et en français. C'est un conte poétique et  
    philosophique sous l'apparence d'un conte pour enfants.",  
    status: 'PUBLISH',  
    authors: ['Antoine de Saint-Exupéry'],  
    language: 'french'  
})
```

← 언어를 'french'(프랑스어)로
지정

9.5 텍스트 검색 언어

- `find()` 또는 `aggregate()` 함수에서 텍스트 검색을 수행할 때:
검색이 사용하는 언어를 MongoDB에 알릴 수 있다

```
> db.books.find({$text: {$search:  
    'simultanment', $language:'french'}}, {title:1}) ← 언어는 프랑스어; 오직 'Le Petit Prince  
{ "_id" : 999, "title" : "Le Petit Prince" }  
  
> db.books.find({$text: {$search: 'simultanment'}}, {title:1}) ← 영어로 동일한 검색을 수행할 경우  
{ "_id" : 186, "title" : "Hadoop in Action" }  
{ "_id" : 293, "title" : "Making Sense of Java" } ← 두 가지 다른 책을 발견하게 된다  
{ "_id" : 999, "title" : "Le Petite Prince" }  
  
> db.books.find({$text: {$search: 'prince'}}, {title:1}) ← 영어로 prince를 검색하면 프랑스어와  
{ "_id" : 145, "title" : "Azure in Action" }  
{ "_id" : 999, "title" : "Le Petit Prince" } ← 영어로 된 책을 찾을 수 있다
```

지정된 언어가 아닌 도큐먼트를 무시한다

9.5 텍스트 검색 언어

- 사용 가능한 언어

Danish, dutch, english, finnish, french, german, hungarian, italian, norwegian, portuguese, romanian, russian, spanish, swedish, turkish, Arabic, dari, iranian persian, urdu, simplified chinese or hans, traditional chinese or hant

none: 형태소 분석과 불용어를 위한 모든 처리를 건너뜀