Multilingual search

Elasticsearch tokeniser

NATTHA WARAPASAKUL

Agenda

- My story
- Why Elasticsearch?
- Text analysis in Elasticsearch
- Tokens
- Language challenge

Agenda

Agenda

My story

Why Elasticsearch?

Text analysis in Elasticsearch

Tokens

Language challenge

Agenda

My story

Why Elasticsearch?

Text analysis in Elasticsearch

Tokens

Language challenge

Agenda

My story

Why Elasticsearch?

Text analysis in Elasticsearch

Tokens

Language challenge

Agenda

- My story
- Why Elasticsearch?
- Text analysis in Elasticsearch
- Tokens
- Language challenge

Many customers prefer to use their own language.

I WAS ASSIGNED TO A SEARCH TEAM.

Is it possible for me to create the search system that support many languages when I can speak only Thai and English

arabic, armenian, basque, bengali, brazilian, bulgarian, catalan, czech, danish, dutch, english, estonian, finnish, french, galician, german, greek, hindi, hungarian, indonesian, irish, italian, latvian, lithuanian, norwegian, persian, portuguese, romanian, russian, sorani, spanish, swedish, turkish, thai.

Agenda

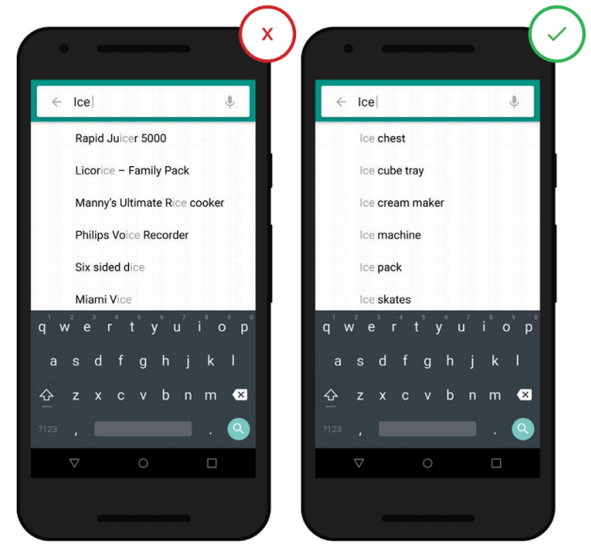
We don't want

- Result is not relevant

- Bad performance for Big data

- Expensive operation

- Go through all data



X Ineffective search indexing delivers a poor search experience.

✓ High-quality indexing gets users targeted, effective results.

Source: https://www.thinkwithgoogle.com/marketing-resources/experience-design/chapter-2-in-app-search/

We want
- R*elevant* results base on matching score

- Boost the result base on what matter to the user.

- Good Performance

- Scale ..., etc

Users type from the beginning of the word

WE WANT TO SUPPORT...

Search terms

Syd
syd
Oper
Opera House
Opera House Sydn

- Partial word search.
- Swap the words.

→

Return results

Sydney Opera House

Agenda

My story

Why Elasticsearch?

Text analysis in Elasticsearch

Tokens

Language challenge

Text analysis

Tokenizer

Analyzer

Tokenization

breaking a text down into smaller chunks, called *tokens*.

Edge-ngram tokenizer

1. Breaks text down into words
2. Remove special character
3. N-gram break the word

Example: **Sydney Opera House.**
[S, Sy, Syd, Sydn, Sydne, Sydney]
[O, Op, Ope, Oper, Opera]
[H, Ho, Hou, Hous, House]

=> 16 tokens

Text analysis

Tokenizer

Analyzer

More Tokenizers

Tokenizer reference | Elasticse

Secure https://www.elastic.co/guide/en/elasticsearch/reference/current/analysis-tokenizers.html#_partial_word_t

Partial Word Tokenizers

edit

These tokenizers break up text or words into small fragments, for partial word matching:

N-Gram Tokenizer

The `ngram` tokenizer can break up text into words when it encounters any of a list of specified characters (e.g. whitespace or punctuation), then it returns n-grams of each word: a sliding window of continuous letters, e.g. `quick` → `[qu, ui, ic, ck]`.

Edge N-Gram Tokenizer

The `edge_ngram` tokenizer can break up text into words when it encounters any of a list of specified characters (e.g. whitespace or punctuation), then it returns n-grams of each word which are anchored to the start of the word, e.g. `quick` → `[q, qu, qui, quic, quick]`.

Structured Text Tokenizers

edit

The following tokenizers are usually used with structured text like identifiers, email addresses, zip codes, and paths, rather than with full text:

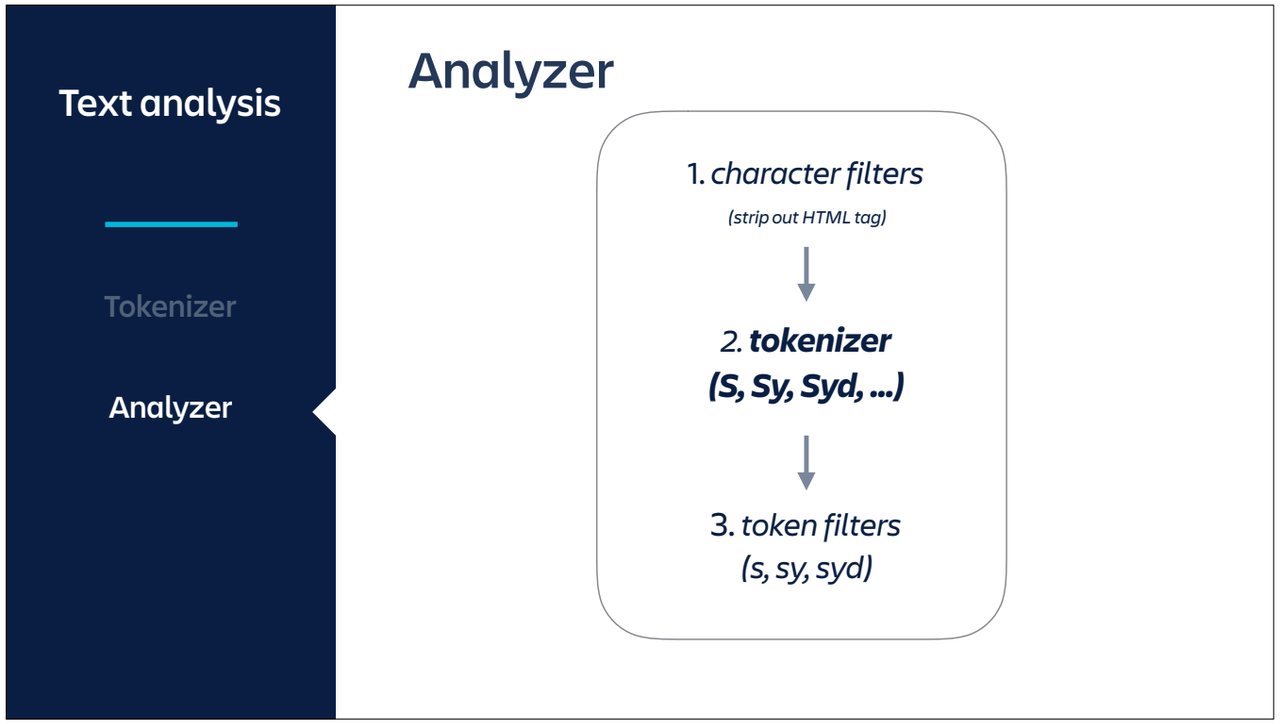
Keyword Tokenizer

The `keyword` tokenizer is a "noop" tokenizer that accepts whatever text it is given and outputs the exact same text as a single term. It can be combined with token filters like `lowercase` to normalise the analysed terms.

Pattern Tokenizer

The `pattern` tokenizer uses a regular expression to either split text into terms whenever it matches a word separator, or to capture matching text as terms.

Ref: https://bit.ly/2wHuq4e



Tokeniser is used as a part of analyzer
An analyzer must have exactly one tokenizer.

Text analysis

Tokenizer

Analyzer

MORE BUILD-IN ANALYSERS

Built-in analyzer reference | El

Secure https://www.elastic.co/guide/en/elasticsearch/reference/current/analysis-analyze

elastic Products Learn Company Pricing Contact

Docs

Elasticsearch Reference [7.6] » Text analysis » Built-in analyzer reference

« Specify an analyzer Fingerprint Analyzer »

Built-in analyzer reference

edit

Elasticsearch ships with a wide range of built-in analyzers, which can be used in any index without further configuration:

Standard Analyzer

The `standard` analyzer divides text into terms on word boundaries, as defined by the Unicode Text Segmentation algorithm. It removes most punctuation, lowercases terms, and supports removing stop words.

Simple Analyzer

The `simple` analyzer divides text into terms whenever it encounters a character which is not a letter. It lowercases all terms.

Whitespace Analyzer

The `whitespace` analyzer divides text into terms whenever it encounters any whitespace character. It does not lowercase terms.

Stop Analyzer

ref: https://bit.ly/32iLIAo

Agenda

My story

Why Elasticsearch?

Text analysis in Elasticsearch

Tokens

Language challenge

Tokens

token is a key to success

Index time

KEY (TOKENS)

Edge-ngram tokeniser + filer

[s, sy, **syd**, sydn, sydne, sydney]
[o, op, ope, ..., opera]
[h, ho, ..., house]

16 tokens

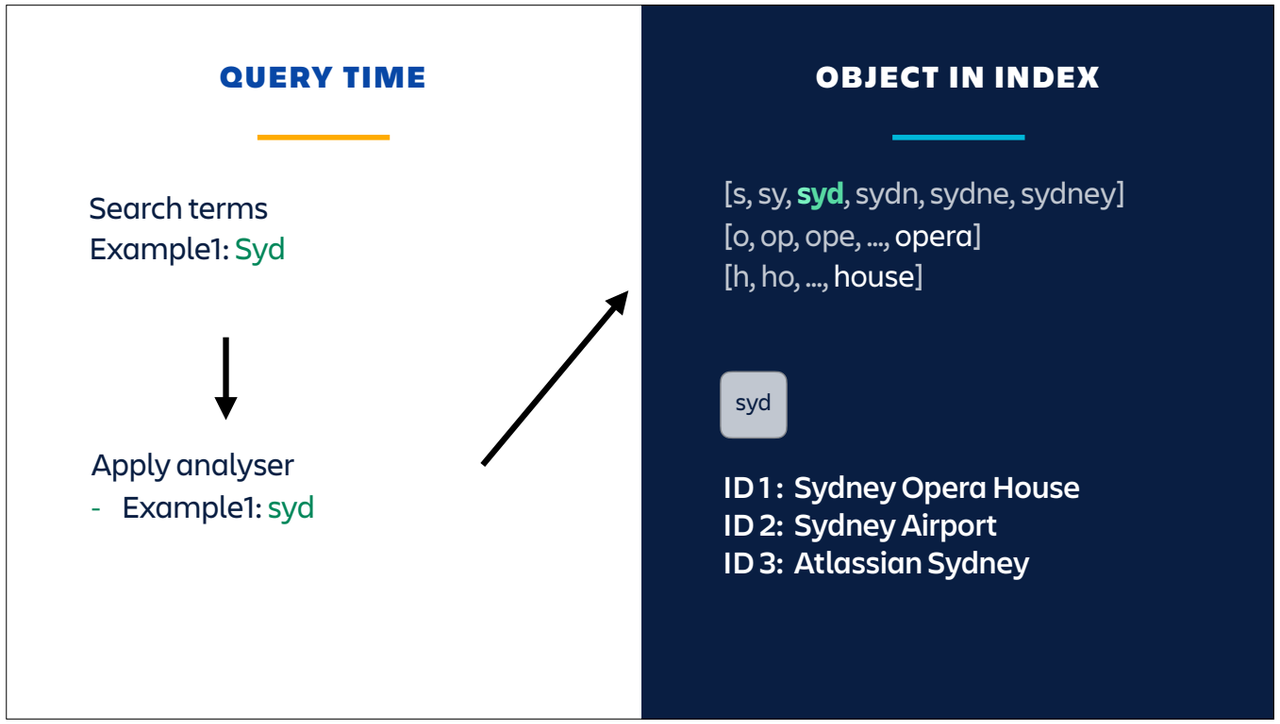
s

sy

syd

VALUE

Json Document {
ObjectID 1 :
Sydney Opera House
}


QUERY TIME
Search terms
Example1: Syd
Apply analyser
- Example1: syd
OBJECT IN INDEX
[s, sy, syd, sydn, sydne, sydney]
[o, op, ope, ..., opera]
[h, ho, ..., house]
syd
ID 1 : Sydney Opera House
ID 2: Sydney Airport
ID 3: Atlassian Sydney

QUERY TIME

Search terms
Example2: Opera house

Apply analyser
- Example2: [“opera”, “house”]

OBJECT IN INDEX

[s, sy, syd, sydn, sydne, sydney]
[o, op, ope, ..., **opera**]
[h, ho, ..., **house**]

opera || house

ID 1 : Sydney Opera House
ID 2: White House
ID 3: Opera concert

Agenda

My story

Why Elasticsearch?

Text analysis in Elasticsearch

Tokens

Language challenge

TRANSLATED DATA FOR “SYDNEY OPERA HOUSE”

Object ID	English	Thai	Arabic	Chinese
1	Sydney Opera House	ซิดนีย์โอเปราเฮาส์	دار سيدني للأوبرا	悉尼歌剧院

Languages

Thai

Arabic

More

No space between words

ซิดนีย์โอเปราเฮาส์ [Sydney Opera House]

[ซ, ซิ, ..., ซิดนีย์]
[โ, โอ, โอเ, ..., โอเปรา]
[เ, เฮ, ..., เฮาส์]

18 tokens from Thai

โอเป => Ope
โอเปราเฮาส์ซิดนีย์ => Opera House Sydney

Languages

Thai

Arabic

More

No space between words
Right to left

دارسيدنيللأوبرا [Sydney Opera House]

دار سيدني للأوبرا

[د,دا, دار]

[سيدني, ...,سي,س]

ل, لل, للأ, للأو,للأوب, للأوبر, للأوبرا

15 tokens from Arabic

دار سيدني للأوبرا
دار الأوبرا

Languages

Thai

Arabic

More

Get ready to be lucky

Languages

Thai

Arabic

More

Language Analysers in Elasticsearch

Language Analyzers | Elasticse ×

Secure | https://www.elastic.co/guide/en/elasticsearch/reference/current/analysis-lang-a

Docs

Elasticsearch Reference [7.6] » Text analysis » Built-in analyzer reference » Language Analyzers

« Keyword Analyzer Pattern Analyzer »

Language Analyzers

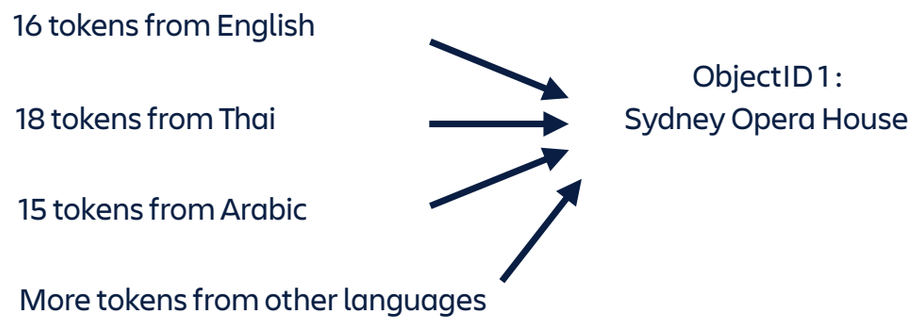
edit

A set of analyzers aimed at analyzing specific language text. The following types are supported: `arabic`, `armenian`, `basque`, `bengali`, `brazilian`, `bulgarian`, `catalan`, `cjk`, `czech`, `danish`, `dutch`, `english`, `estonian`, `finnish`, `french`, `galician`, `german`, `greek`, `hindi`, `hungarian`, `indonesian`, `irish`, `italian`, `latvian`, `lithuanian`, `norwegian`, `persian`, `portuguese`, `romanian`, `russian`, `sorani`, `spanish`, `swedish`, `turkish`, `thai`.

Analysis plugins

- Pinyin
- Japanese analyzer(Kuromoji analyzer)
- etc

Index time


16 tokens from English
18 tokens from Thai
15 tokens from Arabic
More tokens from other languages
ObjectID 1 :
Sydney Opera House

Agenda

My story

Why Elasticsearch?

Text analysis in Elasticsearch

Tokens

Language challenge

Make it more advanced

Make it more advanced

tokens of "**DPS**"

tokens of "**Bali**"

tokens of "**Bali Airport**"

ObjectID 1 :

Denpasar airport

Name : Denpasar Airport
synonyms : DPS, Bali, Bali Airport

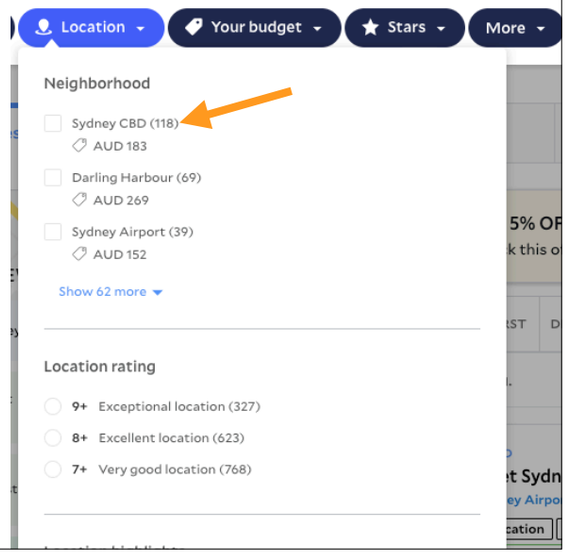
Boost the score
Make **name** match higher than **synonym**

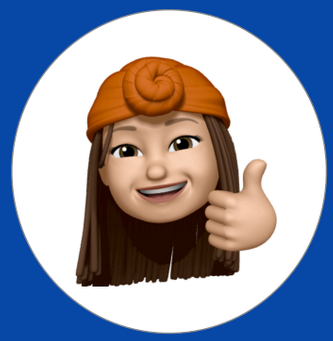
More features to explore

★ Misspelling

★ Filter

★ Aggregation





Thank you

NATTHA WARAPASAKUL