

Full-Text **Search** Internals

Philipp Krenn

@xeraa

Who is using databases?

Who is using search?





elasticsearch



Developer 🥑

Store

Apache Lucene Elasticsearch



elastic cloud

<https://cloud.elastic.co>



docker

```
---
version: '2'
services:

  elasticsearch:
    image: docker.elastic.co/elasticsearch/elasticsearch:$ELASTIC_VERSION
    environment:
      - bootstrap.memory_lock=true
      - "ES_JAVA_OPTS=-Xms512m -Xmx512m"
      - discovery.type=single-node
    ulimits:
      memlock:
        soft: -1
        hard: -1
    mem_limit: 1g
    volumes:
      - esdata1:/usr/share/elasticsearch/data
    ports:
      - 9200:9200

  kibana:
    image: docker.elastic.co/kibana/kibana:$ELASTIC_VERSION
    links:
      - elasticsearch
    ports:
      - 5601:5601

volumes:
  esdata1:
    driver: local
```




Example

These are **not** the droids you
are looking for.

html_strip **Char Filter**

These are not the droids you are looking for.

standard **Tokenizer**

**These are not the droids you are
looking for**

lowercase **Token Filter**

**these are not the droids you are
looking for**

stop **Token Filter**

droids you looking

snowball **Token Filter**

droid you look

Analyze

```
GET /_analyze
{
  "analyzer": "english",
  "text": "These are not the droids you are looking for."
}
```

```
{
  "tokens": [
    {
      "token": "droid",
      "start_offset": 18,
      "end_offset": 24,
      "type": "<ALPHANUM>",
      "position": 4
    },
    {
      "token": "you",
      "start_offset": 25,
      "end_offset": 28,
      "type": "<ALPHANUM>",
      "position": 5
    },
    ...
  ]
}
```

```
GET /_analyze
{
  "char_filter": [
    "html_strip"
  ],
  "tokenizer": "standard",
  "filter": [
    "lowercase",
    "stop",
    "snowball"
  ],
  "text": "These are <em>not</em> the droids you are looking for."
}
```



```
{
  "tokens": [
    {
      "token": "droid",
      "start_offset": 27,
      "end_offset": 33,
      "type": "<ALPHANUM>",
      "position": 4
    },
    {
      "token": "you",
      "start_offset": 34,
      "end_offset": 37,
      "type": "<ALPHANUM>",
      "position": 5
    },
    ...
  ]
}
```

Stop Words

**a an and are as at be but by for if in into is
it no not of on or such that the their then
there these they this to was will with**

**[https://github.com/apache/lucene-solr/blob/master/lucene/
core/src/java/org/apache/lucene/analysis/standard/
StandardAnalyzer.java#L44-L50](https://github.com/apache/lucene-solr/blob/master/lucene/core/src/java/org/apache/lucene/analysis/standard/StandardAnalyzer.java#L44-L50)**

Always Use Stop Words?

To be, or not to be.

German

Das sind nicht die Droiden, nach denen du suchst.

German

droid den such

German with the English Analyzer

da sind nicht die droiden nach
denen du suchst

German Stop Words

https://github.com/apache/lucene-solr/blob/master/lucene/analysis/common/src/resources/org/apache/lucene/analysis/snowball/german_stop.txt

Language Rules

English: Philipp's → philipp

French: l'église → eglis

German: äußerst → ausserst

Detecting Languages

[https://github.com/spinscale/
elasticsearch-ingest-langdetect](https://github.com/spinscale/elasticsearch-ingest-langdetect)

Languages

Arabic, Armenian, Basque, Brazilian, Bulgarian, Catalan, CJK, Czech, Danish, Dutch, English, Finnish, French, Galician, German, Greek, Hindi, Hungarian, Indonesian, Irish, Italian, Latvian, Lithuanian, Norwegian, Persian, Portuguese, Romanian, Russian, Sorani, Spanish, Swedish, Turkish, Thai

More Language Plugins

Core: ICU (Asian languages), Kuromoji (advanced Japanese), Phonetic, SmartCN, Stempel (better Polish stemming), Ukrainian (stemming)

Community: Hebrew, Vietnamese, Network Address Analysis, String2Integer,...

Lemmatizer Plugin

<https://github.com/vhyza/elasticsearch-analysis-lemmagen>

Free: Bulgarian, Czech, English, Estonian, French, Hungarian, Romanian, Slovak, Resian (SL dialect), Slovene, Ukrainian

Non-Commercial: Farsi / Persian, Macedonian, Polish, Russian, Serbian

Another Example

**Obi-Wan never told you what happened to
your father.**

Another Example

**obi wan never told you what
happen your father**

Another Example

No. I am your father.

Another Example

i am your father

Inverted Index

	ID 1	ID 2	ID 3
am	0	0	1[2]
droid	1[4]	0	0
father	0	1[9]	1[4]
happen	0	1[6]	0
i	0	0	1[1]
look	1[7]	0	0
never	0	1[2]	0
obi	0	1[0]	0
told	0	1[3]	0
wan	0	1[1]	0
what	0	1[5]	0
you	1[5]	1[4]	0
your	0	1[8]	1[3]

To / The Index

```
PUT /starwars
{
  "settings": {
    "analysis": {
      "filter": {
        "my_synonym_filter": {
          "type": "synonym",
          "synonyms": [
            "father,dad",
            "droid => droid,machine"
          ]
        }
      }
    }
  },
}
```

```
"analyzer": {  
  "my_analyzer": {  
    "char_filter": [  
      "html_strip"  
    ],  
    "tokenizer": "standard",  
    "filter": [  
      "lowercase",  
      "stop",  
      "snowball",  
      "my_synonym_filter"  
    ]  
  }  
}  
},
```

```
"mappings": {  
  "properties": {  
    "quote": {  
      "type": "text",  
      "analyzer": "my_analyzer"  
    }  
  }  
}
```

```
PUT /starwars/_doc/1
{
  "quote": "These are <em>not</em> the droids you are looking for."
}
PUT /starwars/_doc/2
{
  "quote": "Obi-Wan never told you what happened to your father."
}
PUT /starwars/_doc/3
{
  "quote": "<b>No</b>. I am your father."
}
```

GET /starwars/_doc/1

GET /starwars/_source/1

Search

```
POST /starwars/_search
{
  "query": {
    "match_all": { }
  }
}
```


GET **VS** POST

```
{
  "took": 1,
  "timed_out": false,
  "_shards": {
    "total": 5,
    "successful": 5,
    "failed": 0
  },
  "hits": {
    "total": 3,
    "max_score": 1,
    "hits": [
      {
        "_index": "starwars",
        "_type": "_doc",
        "_id": "2",
        "_score": 1,
        "_source": {
          "quote": "Obi-Wan never told you what happened to your father."
        }
      },
      ...
    ]
  }
}
```

```
POST /starwars/_search
```

```
{  
  "query": {  
    "match": {  
      "quote": "Droid"  
    }  
  }  
}
```



```
POST /starwars/_search
```

```
{  
  "query": {  
    "match": {  
      "quote": "dad"  
    }  
  }  
}
```


POST /starwars/_explain/0

```
{
  "query": {
    "match": {
      "quote": "dad"
    }
  }
}
```

```
{  
  "_index": "starwars",  
  "_type": "_doc",  
  "_id": "0",  
  "matched": false  
}
```



```
POST /starwars/_doc/1/_explain
```

```
{  
  "query": {  
    "match": {  
      "quote": "dad"  
    }  
  }  
}
```

```
{  
  "_index": "starwars",  
  "_type": "_doc",  
  "_id": "1",  
  "matched": false,  
  "explanation": {  
    "value": 0,  
    "description": "no matching term",  
    "details": []  
  }  
}
```

```
POST /starwars/_doc/2/_explain
```

```
{  
  "query": {  
    "match": {  
      "quote": "dad"  
    }  
  }  
}
```

```
{  
  "_index": "starwars",  
  "_type": "_doc",  
  "_id": "2",  
  "matched": true,  
  "explanation": {  
    ...  
  }  
}
```

```
POST /starwars/_search
{
  "query": {
    "match": {
      "quote": "machine"
    }
  }
}
```

```
{
  "took": 2,
  "timed_out": false,
  "_shards": {
    "total": 1,
    "successful": 1,
    "skipped": 0,
    "failed": 0
  },
  "hits": {
    "total": 1,
    "max_score": 1.2499592,
    "hits": [
      {
        "_index": "starwars",
        "_type": "_doc",
        "_id": "1",
        "_score": 1.2499592,
        "_source": {
          "quote": "These are <em>not</em> the droids you are looking for."
        }
      }
    ]
  }
}
```



```
POST /starwars/_search
```

```
{  
  "query": {  
    "match_phrase": {  
      "quote": "I am your father"  
    }  
  }  
}
```



```
POST /starwars/_search
{
  "query": {
    "match_phrase": {
      "quote": {
        "query": "I am father",
        "slop": 1
      }
    }
  }
}
```



```
POST /starwars/_search
{
  "query": {
    "match_phrase": {
      "quote": {
        "query": "I am not your father",
        "slop": 1
      }
    }
  }
}
```



```
POST /starwars/_search
{
  "query": {
    "match": {
      "quote": {
        "query": "van",
        "fuzziness": "AUTO"
      }
    }
  }
}
```



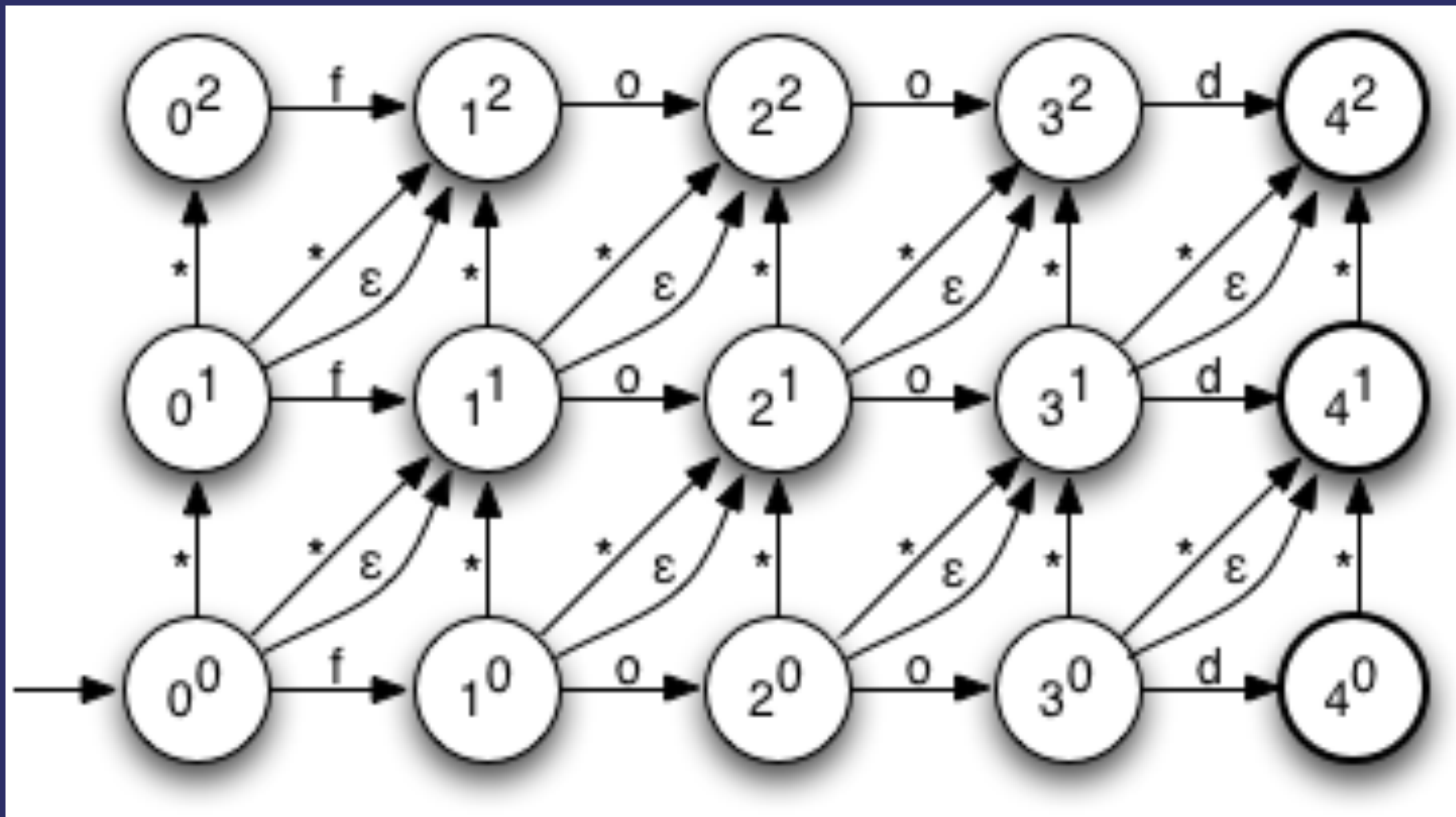
```
POST /starwars/_search
{
  "query": {
    "match": {
      "quote": {
        "query": "ovi-van",
        "fuzziness": 1
      }
    }
  }
}
```


FuzzyQuery History

<http://blog.mikemccandless.com/2011/03/lucenes-fuzzyquery-is-100-times-faster.html>

Before: Brute force

Now: Levenshtein Automaton



<http://blog.notdot.net/2010/07/Damn-Cool-Algorithms-Levenshtein-Automata>

```
SELECT *  
  FROM starwars  
 WHERE quote LIKE "?an" OR  
        quote LIKE "V?n" OR  
        quote LIKE "Va?"
```

Scoring

Term Frequency / Inverse Document Frequency (TF/IDF)

Search one term

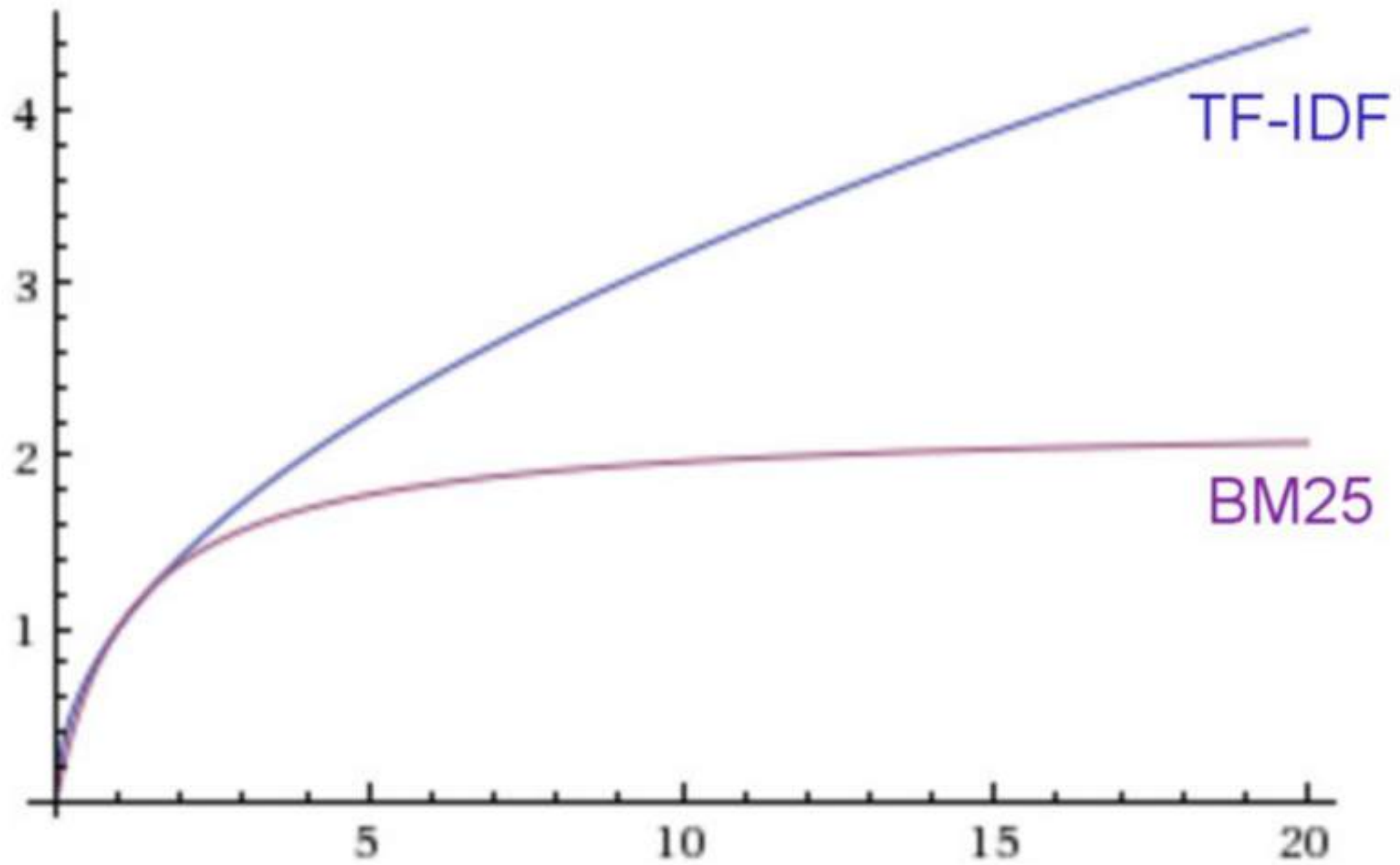
BM25

Default in Elasticsearch 5.0

<https://speakerdeck.com/elastic/improved-text-scoring-with-bm25>

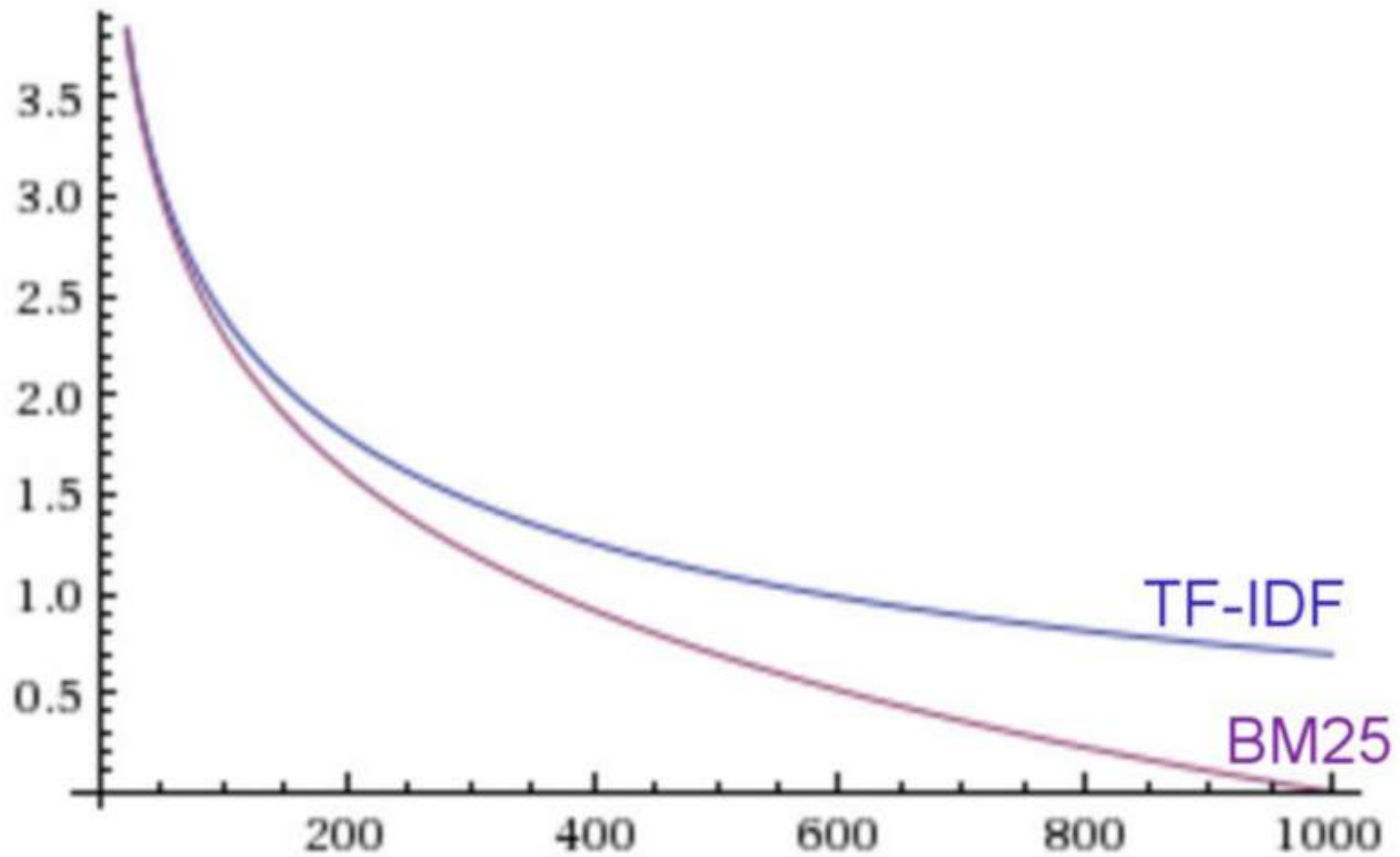
Term Frequency

$$tf(t \text{ in } d) = \sqrt{\text{frequency}}$$



Inverse Document Frequency

$$idf(t) = 1 + \log\left(\frac{numDocs}{docFreq + 1}\right)$$



Field-Length Norm

$$\textit{norm}(d) = \frac{1}{\sqrt{\textit{numTerms}}}$$

```
POST /starwars/_search?explain=true
```

```
{  
  "query": {  
    "match": {  
      "quote": "father"  
    }  
  }  
}
```

```
...
"_explanation": {
  "value": 0.41913947,
  "description": "weight(Synonym(quote:dad quote:father) in 0) [PerFieldSimilarity], result of:",
  "details": [
    {
      "value": 0.41913947,
      "description": "score(doc=0,freq=2.0 = termFreq=2.0\n), product of:",
      "details": [
        {
          "value": 0.2876821,
          "description": "idf(docFreq=1, docCount=1)",
          "details": []
        },
        {
          "value": 1.4569536,
          "description": "tfNorm, computed from:",
          "details": [
            {
              "value": 2,
              "description": "termFreq=2.0",
              "details": []
            }
          ],
          ...
        }
      ]
    }
  ]
}
```

Score

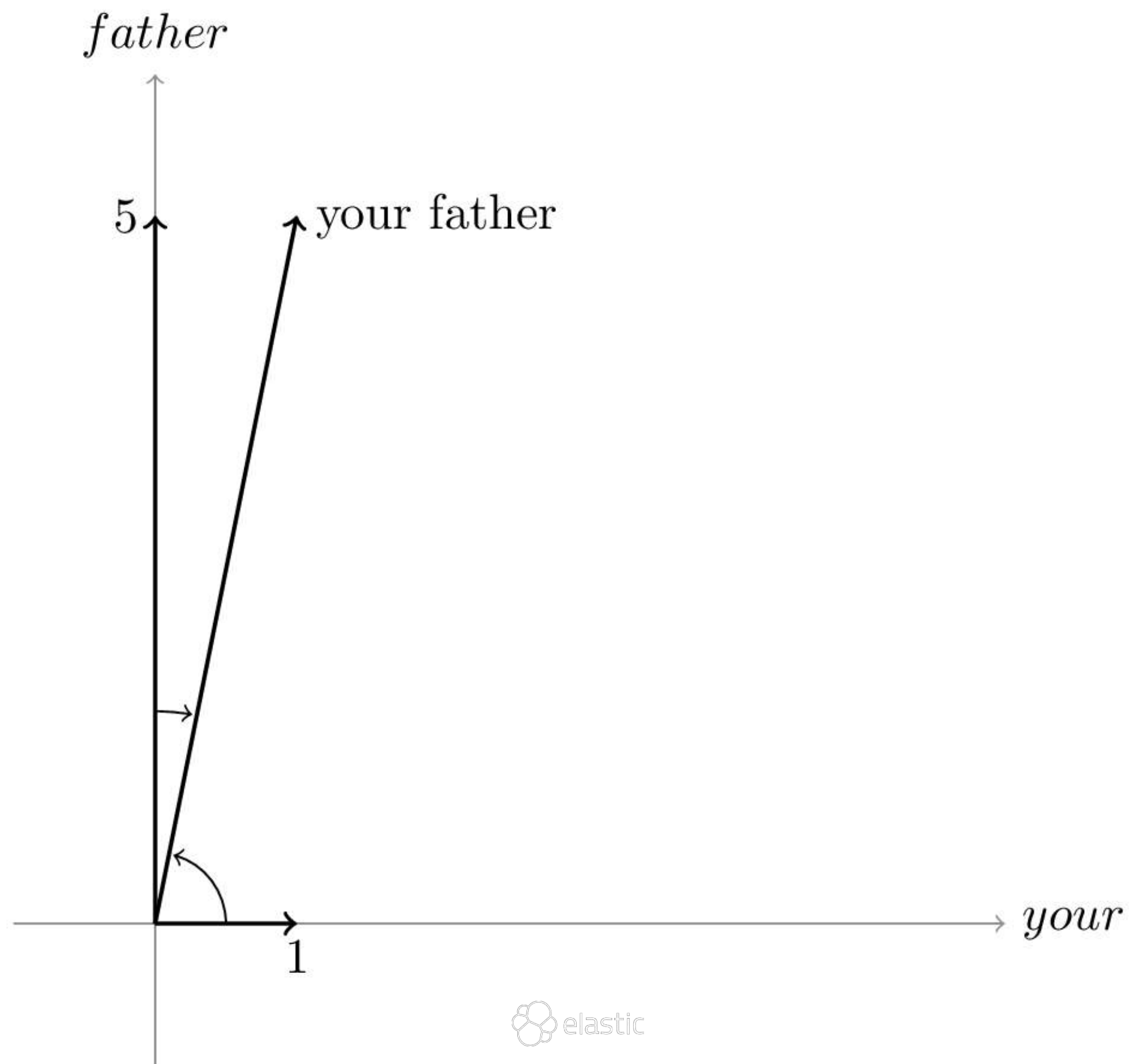
0.41913947: i am your father

**0.39291072: obi wan never told you
what happen your father**

Vector Space Model

Search multiple terms

Search your father



Coordination Factor

Reward multiple terms

Search for 3 terms

1 term: $X \cdot \frac{1}{3}$

2 terms: $(X + Y) \cdot \frac{2}{3}$

3 terms: $(X + Y + Z) \cdot \frac{3}{3}$

Practical Scoring Function

Putting it all together

```
score(q,d) =  
  queryNorm(q)  
  • coord(q,d)  
  •  $\Sigma$  (  
    tf(t in d)  
    • idf(t)2  
    • t.getBoost()  
    • norm(t,d)  
  ) (t in q)
```

Script Score

Function Score (deprecated)

**Script, weight, random, field value, decay
(geo or date)**

POST /starwars/_search

```
{
  "query": {
    "function_score": {
      "query": {
        "match": {
          "quote": "father"
        }
      },
      "random_score": {}
    }
  }
}
```

POST /starwars/_search

```
{
  "size": 1,
  "query": {
    "script_score": {
      "query": {
        "match": {
          "quote": "father"
        }
      },
      "script": {
        "source": "randomScore(1, '_id')"
      }
    }
  }
}
```


Compare Scores

"100% perfect" vs a "50%" match

***Don't do this. Seriously.
Stop trying to think about
your problem this way,
it's not going to end well.***

— [https://cwiki.apache.org/confluence/display/
LUCENEJAVA/ScoresAsPercentages](https://cwiki.apache.org/confluence/display/LUCENEJAVA/ScoresAsPercentages)

```
GET /starwars/_analyze
{
  "analyzer" : "my_analyzer",
  "text": "These are my father's machines."
}
```



```
PUT /starwars/_doc/4
{
  "quote": "These are my father's machines."
}
```

```
POST /starwars/_search
{
  "query": {
    "match": {
      "quote": "my father machine"
    }
  }
}
```



```
"hits": {
  "total": 4,
  "max_score": 2.92523,
  "hits": [
    {
      "_index": "starwars",
      "_type": "_doc",
      "_id": "4",
      "_score": 2.92523,
      "_source": {
        "quote": "These are my father's machines."
      }
    },
    {
      "_index": "starwars",
      "_type": "_doc",
      "_id": "1",
      "_score": 0.8617505,
      "_source": {
        "quote": "These are <em>not</em> the droids you are looking for."
      }
    }
  ],
  ...
}
```

2.92523 == 100%


```
DELETE /starwars/_doc/4
```

```
POST /starwars/_search
```

```
{  
  "query": {  
    "match": {  
      "quote": "my father machine"  
    }  
  }  
}
```

```
"hits": {
  "total": 3,
  "max_score": 1.2499592,
  "hits": [
    {
      "_index": "starwars",
      "_type": "_doc",
      "_id": "1",
      "_score": 1.2499592,
      "_source": {
        "quote": "These are <em>not</em> the droids you are looking for."
      }
    },
    ...
  ]
}
```

1.2499592 == 43%
or 100%?

```
PUT /starwars/_doc/4
{
  "quote": "These droids are my father's father's machines."
}
```

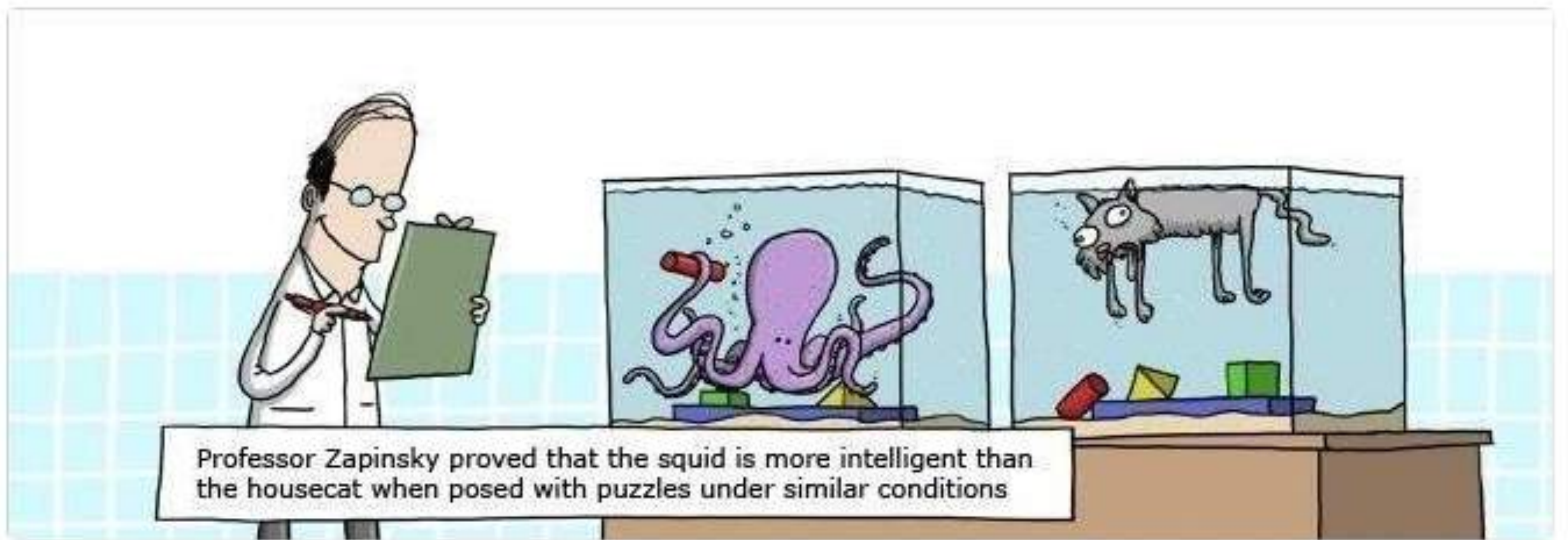
```
POST /starwars/_search
{
  "query": {
    "match": {
      "quote": "my father machine"
    }
  }
}
```

```
"hits": {
  "total": 4,
  "max_score": 3.0068164,
  "hits": [
    {
      "_index": "starwars",
      "_type": "_doc",
      "_id": "4",
      "_score": 3.0068164,
      "_source": {
        "quote": "These droids are my father's father's machines."
      }
    },
    {
      "_index": "starwars",
      "_type": "_doc",
      "_id": "1",
      "_score": 0.89701396,
      "_source": {
        "quote": "These are <em>not</em> the droids you are looking for."
      }
    }
  ],
  ...
}
```

3.0068164 == 103%?



Performance



ONE MILLION



DOCUMENTS

AI / ML / Deep Learning

Everytime someone contacts @o19s and asks for "learning to rank" or some other ML relevance thing often they haven't done step 1 (some basic relevance work)



Emmanuel Ameisen @mlpowered · Jul 16

How to ship ML in practice:

1/ Write a simple rule based solution to cover 80% of use cases

2/ Write a simple ML algorithm to cover 95% of cases

3/ Write a filtering algorithm to route inputs to the correct method

4/ Add monitoring

5/ Detect drift

...

24/ Deep Learning

[Show this thread](#)

<https://twitter.com/softwareDoug/status/1151555116769730560>

Conclusion

Indexing

Formatting

Tokenize

Lowercase, Stop Words, Stemming

Synonyms

Scoring

Term Frequency

Inverse Document Frequency

Field-Length Norm

Vector Space Model

Advanced Queries

Highlighting

(Edge) NGrams & Search as You Type

Multiple Analyzers

Reindex & Alias

There is more
Elastic Stack

Trainings

<https://training.elastic.co>



app search

Find

Burritos

Near

San Francisco, CA

Q

🕒

Open Now

Price

\$

\$\$

\$\$\$

\$\$\$\$

Parking:

●

 Street

●

 Garage

●

 Valet

Showing 1-10 of 100

- ★ ★ ★ ★ ★

burrito
- ★ ★ ★ ★ ★

burritos
- ★ ★ ★ ★ ★

burrito



Search

Time range filter

Structured search

Stemming / Highlighting

Numeric filter

Pagination

Geo shape filter

- Overview
- Analytics
- Query Tester

MANAGE

- Content
 - Fields
 - API Logs
- RELEVANCE
- Weights
 - Boosts

SEARCH SETTINGS

- Pinned Results**
- Synonyms

Result Pins for "sedan"

sedan ☐ full-size [+ Add Alias](#)

Promoted Results

1 ID 710766 vehicle_type ⇒ sedan
model_make_id ⇒ BMW 335i
model_trim ⇒ M Trim
model_drive ⇒ RWD
SCORE 0.991

2 ID 310465 vehicle_type ⇒ sedan
model_make_id ⇒ Toyota
model_trim ⇒ EX
model_drive ⇒ FWD
SCORE 1.359

3 ID 007891 vehicle_type ⇒ sedan
model_make_id ⇒ Toyota
model_trim ⇒ EX
model_drive ⇒ FWD
SCORE 1.143

4 ID 368 vehicle_type ⇒ sedan
model_make_id ⇒ Toyota
model_trim ⇒ EX
model_drive ⇒ FWD
SCORE 1.143

sedan

PROMOTED



BMW 335i

\$126.05 Per day

Dealer

Unlimited Mileage

1.2 mi



Toyota Camry

\$126.05 Per day

Dealer

625 Free Miles

.8 mi



Nissan Altima

\$126.05 Per day

Dealer

Unlimited Mileage

1.4 mi



Lexus IS350

\$126.05 Per day

Dealer

Unlimited Mileage

2.1 mi

Thank You!

Questions?

Philipp Krenn

@xeraa

PS: Stickers

More

```
POST /starwars/_search
{
  "query": {
    "match": {
      "quote": "father"
    }
  },
  "highlight": {
    "type": "unified",
    "pre_tags": [
      "<tag>"
    ],
    "post_tags": [
      "</tag>"
    ],
    "fields": {
      "quote": {}
    }
  }
}
```



```
...
"hits": [
  {
    "_index": "starwars",
    "_type": "_doc",
    "_id": "3",
    "_score": 0.41913947,
    "_source": {
      "quote": "<b>No</b>. I am your father."
    },
    "highlight": {
      "quote": [
        "<b>No</b>. I am your <tag>father</tag>."
      ]
    }
  },
  ...
]
```

Boolean Queries

must must_not should filter

```
POST /starwars/_search
```

```
{
  "query": {
    "bool": {
      "must": {
        "match": {
          "quote": "father"
        }
      },
      "should": [
        {
          "match": {
            "quote": "your"
          }
        },
        {
          "match": {
            "quote": "obi"
          }
        }
      ]
    }
  }
}
```



```
POST /starwars/_search
```

```
{
  "query": {
    "bool": {
      "filter": {
        "match": {
          "quote": "father"
        }
      },
      "should": [
        {
          "match": {
            "quote": "your"
          }
        },
        {
          "match": {
            "quote": "obi"
          }
        }
      ]
    }
  }
}
```


Named Queries & `minimum_should_match`


```
...
  "hits": {
    "total": 1,
    "max_score": 1.8154771,
    "hits": [
      {
        "_index": "starwars",
        "_type": "_doc",
        "_id": "2",
        "_score": 1.8154771,
        "_source": {
          "quote": "Obi-Wan never told you what happened to your father."
        },
        "matched_queries": [
          "quote-obi",
          "quote-your"
        ]
      }
    ]
  }
}
```

Boosting

>1 increase, <1 decrease, ~~≤ 0 punish~~

Suggestion

Suggest a similar text

`_search` **end point**

`_suggest` **deprecated since 5.0**

POST /starwars/_search

```
{
  "query": {
    "match": {
      "quote": "druir"
    }
  },
  "suggest": {
    "my_suggestion" : {
      "text" : "druir",
      "term" : {
        "field" : "quote"
      }
    }
  }
}
```



```
...
  "hits": {
    "total": 0,
    "max_score": null,
    "hits": []
  },
  "suggest": {
    "my_suggestion": [
      {
        "text": "dru",
        "offset": 0,
        "length": 4,
        "options": [
          {
            "text": "droid",
            "score": 0.5,
            "freq": 1
          }
        ]
      }
    ]
  }
}
```

NGram

Partial matches

Trigram & Edge Gram

search_as_you_type


```
GET /_analyze
{
  "char_filter": [
    "html_strip"
  ],
  "tokenizer": {
    "type": "ngram",
    "min_gram": "3",
    "max_gram": "3",
    "token_chars": [
      "letter"
    ]
  },
  "filter": [
    "lowercase"
  ],
  "text": "These are <em>not</em> the droids you are looking for."
}
```



```
GET /_analyze
{
  "char_filter": [
    "html_strip"
  ],
  "tokenizer": {
    "type": "edge_ngram",
    "min_gram": "1",
    "max_gram": "3",
    "token_chars": [
      "letter"
    ]
  },
  "filter": [
    "lowercase"
  ],
  "text": "These are <em>not</em> the droids you are looking for."
}
```


search_as_you_type (7.2)

Combining Analyzers

Reindex

Store multiple times

Combine scores


```
PUT /starwars_v42
{
  "settings": {
    "analysis": {
      "filter": {
        "my_synonym_filter": {
          "type": "synonym",
          "synonyms": [
            "droid,machine",
            "father,dad"
          ]
        },
        "my_ngram_filter": {
          "type": "ngram",
          "min_gram": "3",
          "max_gram": "3",
          "token_chars": [
            "letter"
          ]
        }
      }
    }
  },
}
```

```
"analyzer": {
  "my_lowercase_analyzer": {
    "char_filter": [
      "html_strip"
    ],
    "tokenizer": "whitespace",
    "filter": [
      "lowercase"
    ]
  },
  "my_full_analyzer": {
    "char_filter": [
      "html_strip"
    ],
    "tokenizer": "standard",
    "filter": [
      "lowercase",
      "stop",
      "snowball",
      "my_synonym_filter"
    ]
  },
}
```



```
"my_ngram_analyzer": {  
  "char_filter": [  
    "html_strip"  
  ],  
  "tokenizer": "whitespace",  
  "filter": [  
    "lowercase",  
    "stop",  
    "my_ngram_filter"  
  ]  
}  
}  
},
```

```
"mappings": {  
  "properties": {  
    "quote": {  
      "type": "text",  
      "fields": {  
        "lowercase": {  
          "type": "text",  
          "analyzer": "my_lowercase_analyzer"  
        },  
        "full": {  
          "type": "text",  
          "analyzer": "my_full_analyzer"  
        },  
        "ngram": {  
          "type": "text",  
          "analyzer": "my_ngram_analyzer"  
        }  
      }  
    }  
  }  
}
```

```
POST /_reindex
{
  "source": {
    "index": "starwars"
  },
  "dest": {
    "index": "starwars_v42"
  }
}
```

```
PUT _alias
{
  "actions": [
    {
      "add": {
        "index": "starwars_v42",
        "alias": "starwars_extended"
      }
    }
  ]
}
```

Aliases

Atomic remove **and** add

Point to multiple indices (read-only)

```
POST /starwars_extended/_search?explain=true
{
  "query": {
    "multi_match": {
      "query": "obiwan",
      "fields": [
        "quote",
        "quote.lowercase",
        "quote.full",
        "quote.ngram"
      ],
      "type": "most_fields"
    }
  }
}
```



```
...
"hits": {
  "total": 1,
  "max_score": 0.4912064,
  "hits": [
    {
      "_shard": "[starwars_v42][2]",
      "_node": "BCDwzJ4WSw2dyoGLTzw1qw",
      "_index": "starwars_v42",
      "_type": "_doc",
      "_id": "2",
      "_score": 0.4912064,
      "_source": {
        "quote": "Obi-Wan never told you what happened to your father."
      },
    },
  ],
  ...
}
```

Whitespace Tokenizer

```
"weight(  
  Synonym(quote.ngram:biw quote.ngram:iwa quote.ngram:obi quote.ngram:wan)  
in 0) [PerFieldSimilarity], result of:"
```



```
POST /starwars_extended/_search
{
  "query": {
    "multi_match": {
      "query": "you",
      "fields": [
        "quote",
        "quote.lowercase",
        "quote.full^5",
        "quote.ngram"
      ],
      "type": "best_fields"
    }
  }
}
```


Multi Match Type

`best_fields` **Score of the best field (default)**

`cross_fields` **All terms in at least one field**

`most_fields` **Score sum of all fields**

phrase

Different Analyzers for Indexing and Searching

Per query

In the mapping

```
POST /starwars_extended/_search
{
  "query": {
    "match": {
      "quote.ngram": {
        "query": "the",
        "analyzer": "standard"
      }
    }
  }
}
```

```
...
"hits": [
  {
    "_index": "starwars_extended",
    "_type": "_doc",
    "_id": "2",
    "_score": 0.38254172,
    "_source": {
      "quote": "Obi-Wan never told you what happened to your father."
    }
  },
  {
    "_index": "starwars_extended",
    "_type": "_doc",
    "_id": "3",
    "_score": 0.36165747,
    "_source": {
      "quote": "<b>No</b>. I am your father."
    }
  }
]
...
```


Edge Gram vs Trigram

Extending a mapping

Testing a custom mapping

```
POST /starwars_extended/_close
PUT /starwars_extended/_settings
{
  "analysis": {
    "filter": {
      "my_edgegram_filter": {
        "type": "edge_ngram",
        "min_gram": 3,
        "max_gram": 10
      }
    },
    "analyzer": {
      "my_edgegram_analyzer": {
        "char_filter": [
          "html_strip"
        ],
        "tokenizer": "standard",
        "filter": [
          "lowercase",
          "my_edgegram_filter"
        ]
      }
    }
  }
}
POST /starwars_extended/_open
```



```
GET starwars_extended/_analyze
{
  "text": "Father",
  "analyzer": "my_edgegram_analyzer"
}
```



```
PUT /starwars_extended/_mapping
{
  "properties": {
    "quote": {
      "type": "text",
      "fields": {
        "edgegram": {
          "type": "text",
          "analyzer": "my_edgegram_analyzer",
          "search_analyzer": "standard"
        }
      }
    }
  }
}
```

```
PUT /starwars_extended/_doc/4
{
  "quote": "I find your lack of faith disturbing."
}
PUT /starwars_extended/_doc/5
{
  "quote": "That... is your failure."
}
```

```
GET /starwars_extended/_termvectors/4
{
  "fields": [
    "quote.edgogram"
  ],
  "offsets": true,
  "payloads": true,
  "positions": true,
  "term_statistics": true,
  "field_statistics": true
}
```

```
{
  "_index": "starwars_v42",
  "_type": "_doc",
  "_id": "4",
  "_version": 1,
  "found": true,
  "took": 3,
  "term_vectors": {
    "quote.edgegram": {
      "field_statistics": {
        "sum_doc_freq": 26,
        "doc_count": 2,
        "sum_ttf": 26
      },
      "terms": {
        "dis": {
          "doc_freq": 1,
          "ttf": 1,
          "term_freq": 1,
          "tokens": [
            {
              "position": 6,
              "start_offset": 26,
              "end_offset": 36
            }
          ]
        }
      },
      "dist": {
        "doc_freq": 1,
        "ttf": 1,
        ...
      }
    }
  }
}
```


POST /starwars_extended/_search

```
{  
  "query": {  
    "match": {  
      "quote": "fail"  
    }  
  }  
}
```

```
POST /starwars_extended/_search
{
  "query": {
    "match": {
      "quote.lowercase": "fail"
    }
  }
}
```



```
POST /starwars_extended/_search
{
  "query": {
    "match": {
      "quote.full": "fail"
    }
  }
}
```

```
POST /starwars_extended/_search
{
  "query": {
    "match": {
      "quote.ngram": "fail"
    }
  }
}
```



```
POST /starwars_extended/_search
{
  "query": {
    "match": {
      "quote.edgegram": "fail"
    }
  }
}
```

```
...
"hits": {
  "total": 1,
  "max_score": 0.39556286,
  "hits": [
    {
      "_index": "starwars_v42",
      "_type": "_doc",
      "_id": "5",
      "_score": 0.39556286,
      "_source": {
        "quote": "That... is your failure."
      }
    }
  ]
}
...
```