Multilingual Search Support in European E-commerce: A Journey with Apache Lucene

Haystack Europe
2023-09-20
@lucianprecup
@a2lean

Haystack On Tour Paris - november 2022

2024 - the year of ...

- 2022 - Vector Search
- 2023 - Large Language Models
- 2024 - ...

2024 - the year of ...

- 2022 - Vector Search
- 2023 - Large Language Models
- 2024 - ... Apache Lucene?

Apache Lucene - 22 years and counting

Apache Lucene - 22 years and counting

Doug Cutting
@cutting

Lucene's FuzzyQuery is 100 times faster!
blog.mikemccandless.com/2011/03/lucene...

9:58 PM · Mar 24, 2011

Uwe Schindler
@thetaph1

Mike McCandless talks about #Apache #Lucene that helps to squash #Java #JVM bugs: elastic.co/blog/lucene-jv...

9:45 PM · Jul 17, 2015 from Bremen, Germany

Adrien Grand
@jpountz

I ran some benchmarks between Lucene 9.7 and 9.8 (soon to be released), as well as with recursive graph partitioning enabled (-bp): jpountz.github.io/lucene-9.7-vs-... There's a nice speedup on 9.8 alone, and then recursive graph bisection gives another great speedup.

11:46 PM · Sep 13, 2023 · **3,107** Views

Uwe Schindler
@thetaph1

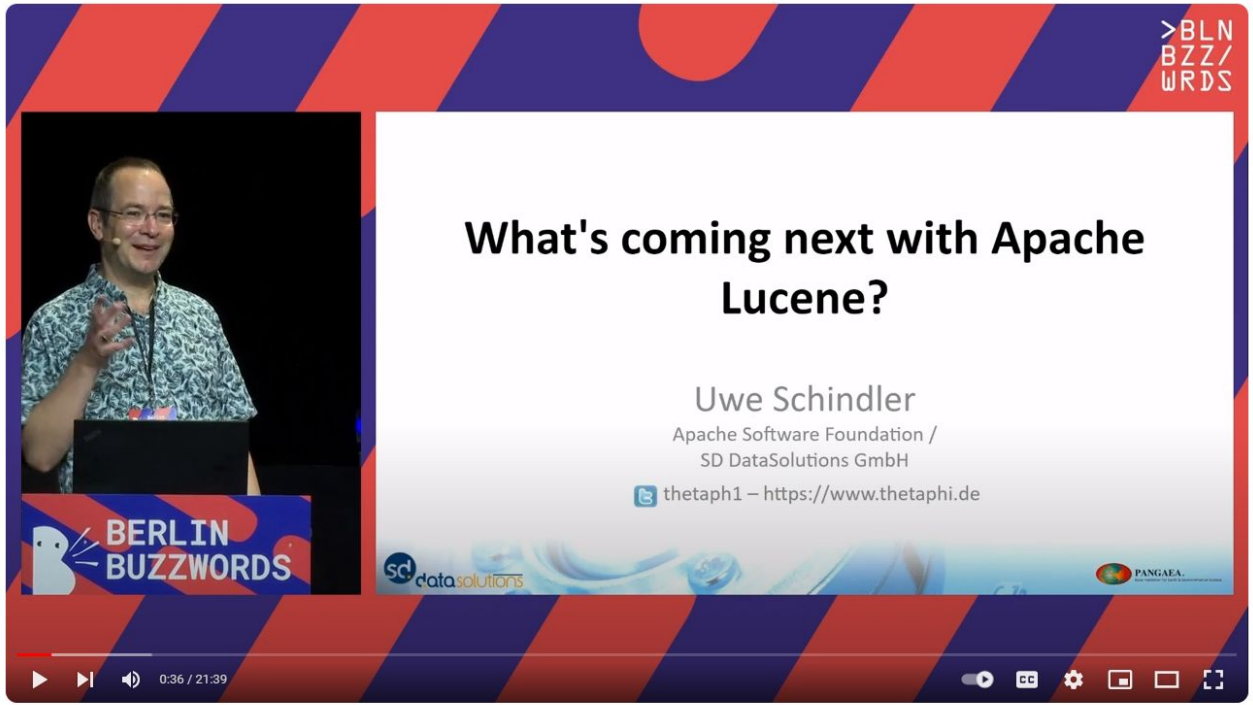
#Apache #Lucene can much faster execute kNN vector queries by calculating dot products / cosine distances using SIMD instructions on AVX2 (x86) and NEON (ARM). It will only work with #Java20 on coming Lucene 9.7 with "--add-modules jdk.incubator.vector":

apache/lucene
#12311 **Integrate the Incubating Panama Vector API**
165 comments · 62 reviews · 16 files · +1025 −179
ChrisHegarty · May 18, 2023 · 59 commits

github.com
Integrate the Incubating Panama Vector API by ChrisHegarty · Pull Request #...
Leverage accelerated vector hardware instructions in Vector Search. Lucene already has a mechanism that enables the use of non-final JDK APIs, currentl...

7:26 PM · May 26, 2023 · **23.1K** Views

2031 at Berlin Buzzwords



Thanks to Apache Lucene

Apache Nutch – provides web crawling and HTML parsing

Apache Solr – an enterprise search server

Elasticsearch – an enterprise search server released in 2010

MongoDB Atlas Search – a cloud-native enterprise search application based on MongoDB and Apache Lucene

OpenSearch – an open source enterprise search server based on a fork of Elasticsearch 7

Adelean a2 - an e-commerce and community search server

Who am I ?

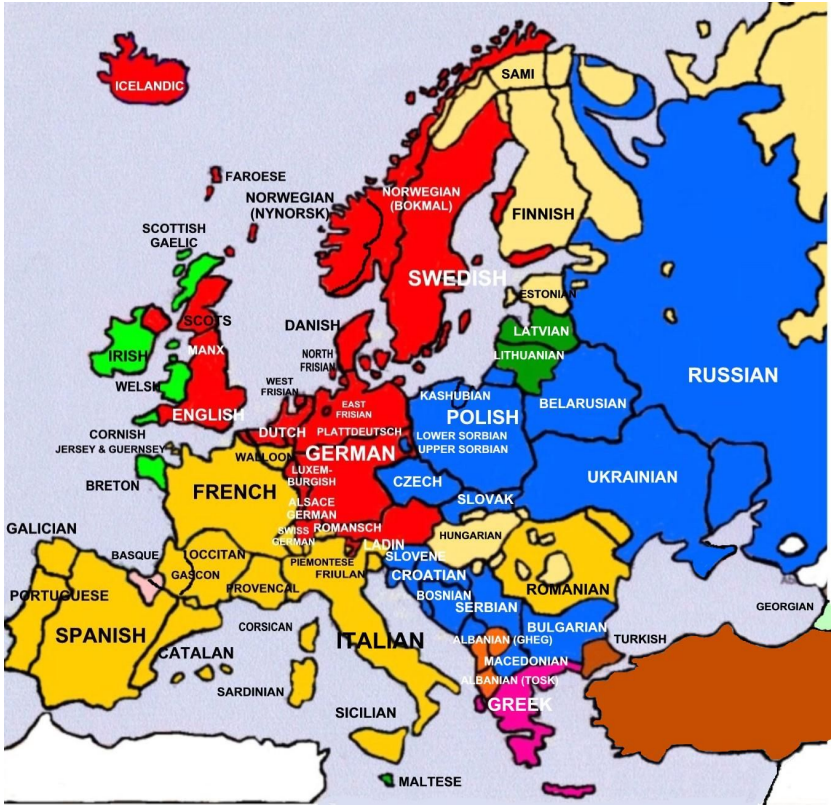
Adelean

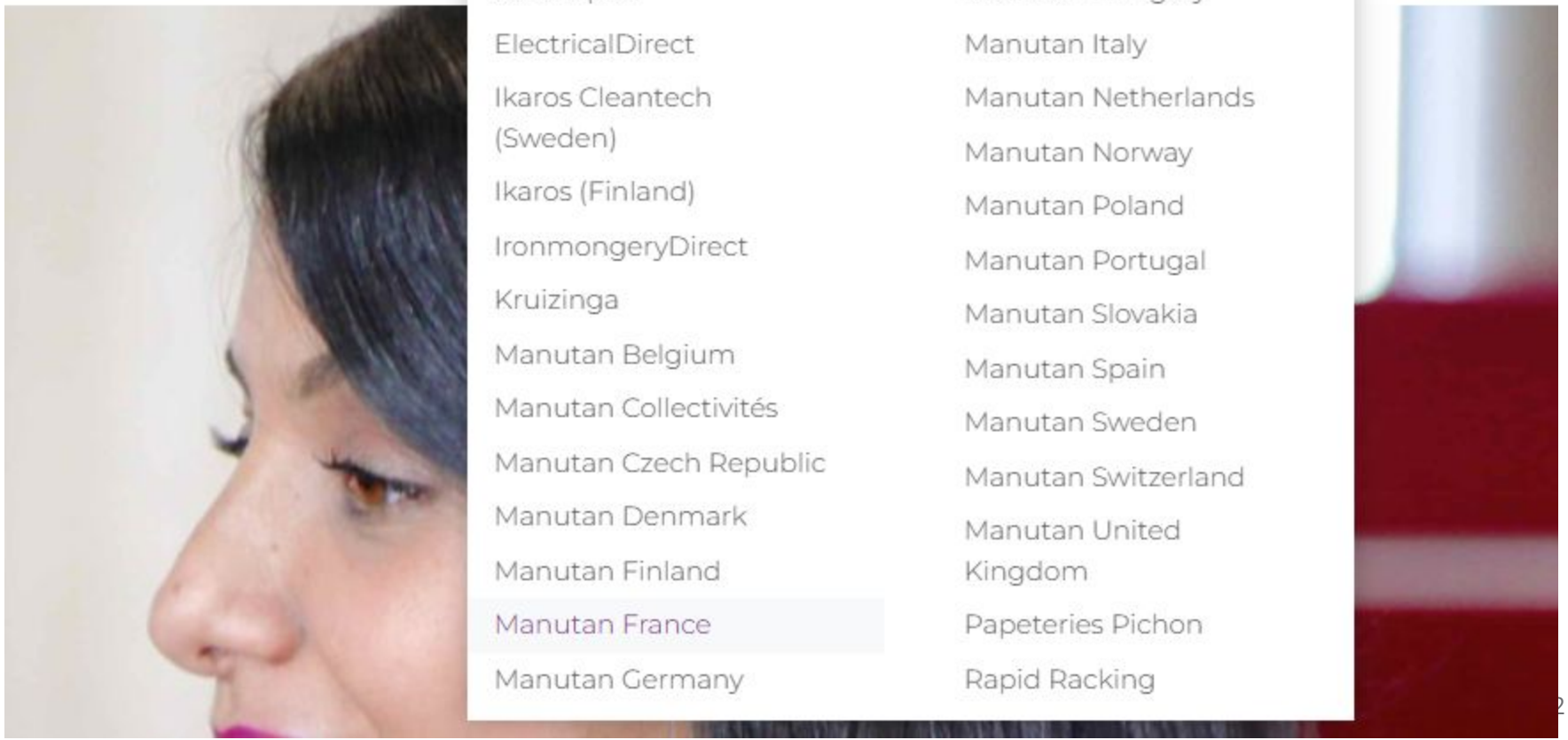
- Experts in **Search** technologies
- Integrators of **Elasticsearch, OpenSearch** and **Solr**
- **Consulting** and **Training** providers
- Developers of **a2** - **E-commerce** and **Enterprise Search** solution
  Developpers of **all.site** - your **Collaborative** Search Engine

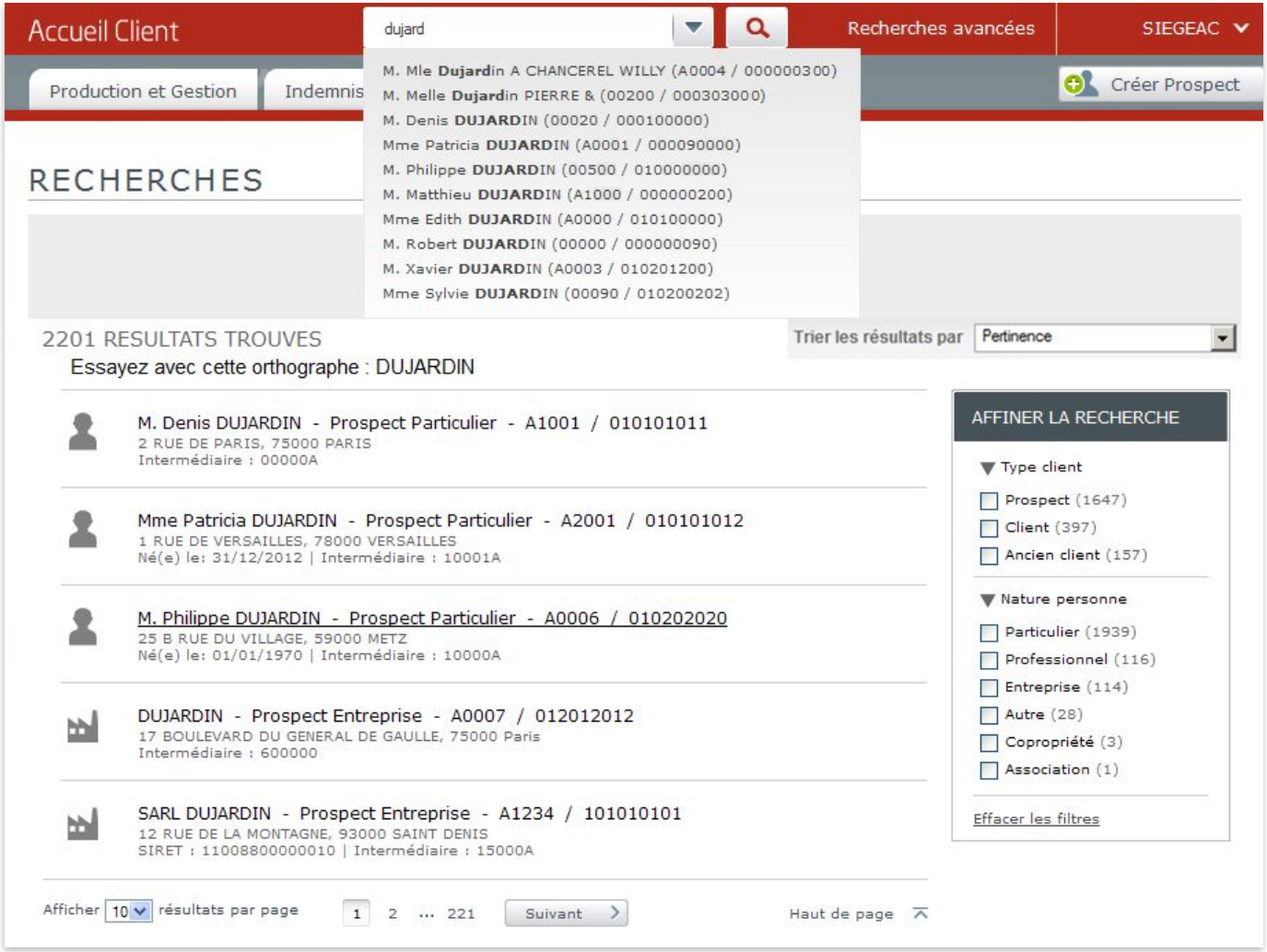
Multilingual Search Support in European E-commerce: A Journey with Apache Lucene

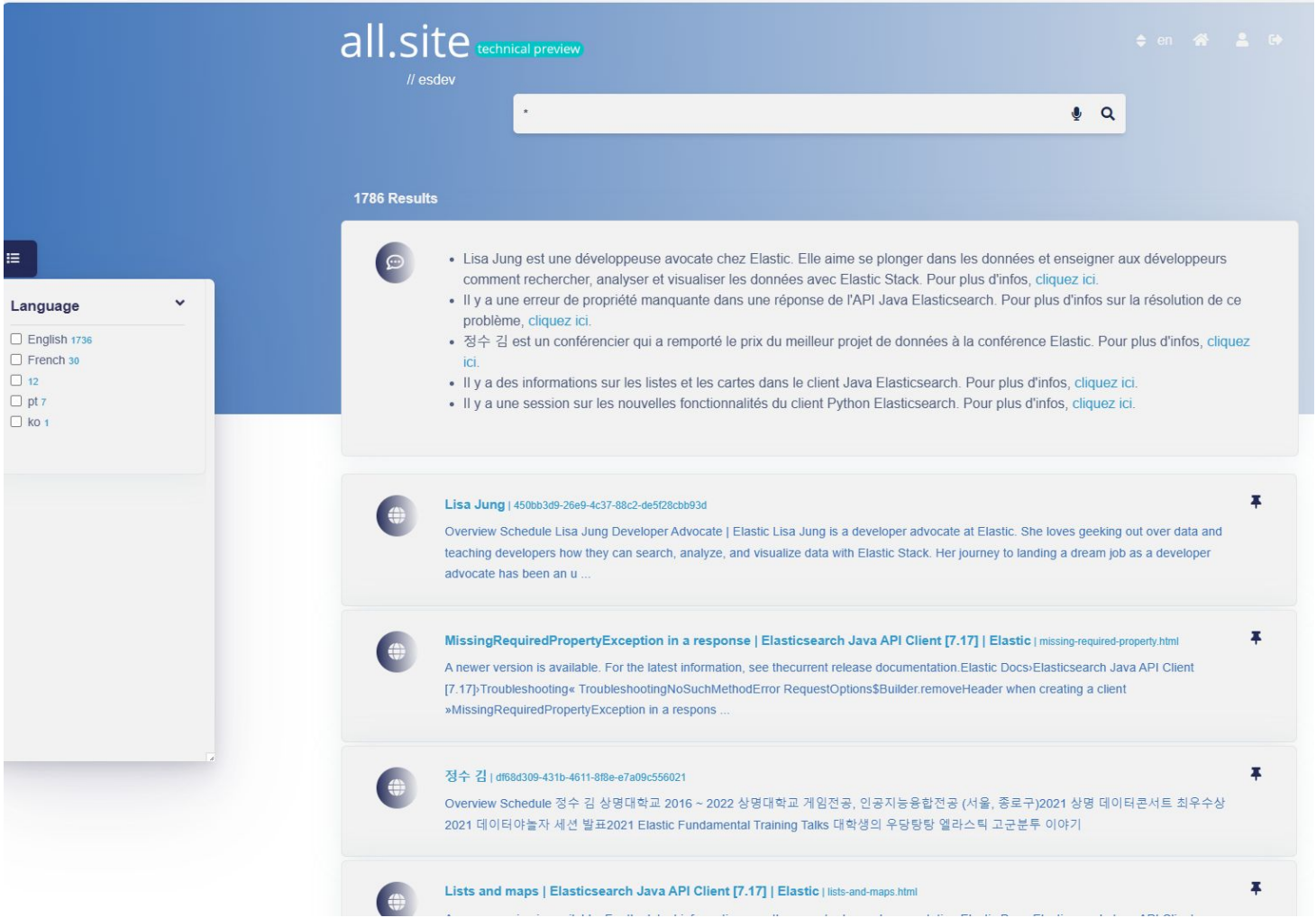
- Lucene levers for processing linguistic specificities
- Tips, tricks, caveats and hacks
- E-commerce specific use cases
- Dealing with person names specific use cases
- Language specific libraries and optimizations
- Getting things done in production today

European languages




OUR ACTIVITY
OUR VISION
JOIN US
PRESS
OUR E-COMMERCE WEBSITES
Casal Sport
ElectricalDirect
Ikaros Cleantech (Sweden)
Ikaros (Finland)
IronmongeryDirect
Kruizinga
Manutan Belgium
Manutan Collectivités
Manutan Czech Republic
Manutan Denmark
Manutan Finland
Manutan France
Manutan Germany
Manutan Hungary
Manutan Italy
Manutan Netherlands
Manutan Norway
Manutan Poland
Manutan Portugal
Manutan Slovakia
Manutan Spain
Manutan Sweden
Manutan Switzerland
Manutan United Kingdom
Papeteries Pichon
Rapid Racking


Accueil Client
dujard
Recherches avancées
SIEGEAC
Production et Gestion
Indemnis
Créer Prospect
M. Mle Dujardin A CHANCEREL WILLY (A0004 / 000000300)
M. Melle Dujardin PIERRE & (00200 / 000303000)
M. Denis DUJARDIN (00020 / 000100000)
Mme Patricia DUJARDIN (A0001 / 000090000)
M. Philippe DUJARDIN (00500 / 010000000)
M. Matthieu DUJARDIN (A1000 / 000000200)
Mme Edith DUJARDIN (A0000 / 010100000)
M. Robert DUJARDIN (00000 / 000000090)
M. Xavier DUJARDIN (A0003 / 010201200)
Mme Sylvie DUJARDIN (00090 / 010200202)
RECHERCHES
2201 RESULTATS TROUVES
Trier les résultats par
Pertinence
Essayez avec cette orthographe : DUJARDIN
M. Denis DUJARDIN - Prospect Particulier - A1001 / 010101011
2 RUE DE PARIS, 75000 PARIS
Intermédiaire : 00000A
Mme Patricia DUJARDIN - Prospect Particulier - A2001 / 010101012
1 RUE DE VERSAILLES, 78000 VERSAILLES
Né(e) le: 31/12/2012 | Intermédiaire : 10001A
M. Philippe DUJARDIN - Prospect Particulier - A0006 / 010202020
25 B RUE DU VILLAGE, 59000 METZ
Né(e) le: 01/01/1970 | Intermédiaire : 10000A
DUJARDIN - Prospect Entreprise - A0007 / 012012012
17 BOULEVARD DU GENERAL DE GAULLE, 75000 Paris
Intermédiaire : 600000
SARL DUJARDIN - Prospect Entreprise - A1234 / 101010101
12 RUE DE LA MONTAGNE, 93000 SAINT DENIS
SIRET : 11008800000010 | Intermédiaire : 15000A
AFFINER LA RECHERCHE
Type client
Prospect (1647)
Client (397)
Ancien client (157)
Nature personne
Particulier (1939)
Professionnel (116)
Entreprise (114)
Autre (28)
Copropriété (3)
Association (1)
Effacer les filtres
Afficher 10 résultats par page
1 2 ... 221
Suivant
Haut de page


all.site
technical preview
// esdev
en
*
1786 Results
Language
English 1736
French 30
12
pt 7
ko 1
Lisa Jung est une développeuse avocate chez Elastic. Elle aime se plonger dans les données et enseigner aux développeurs comment rechercher, analyser et visualiser les données avec Elastic Stack. Pour plus d'infos, cliquez ici.
Il y a une erreur de propriété manquante dans une réponse de l'API Java Elasticsearch. Pour plus d'infos sur la résolution de ce problème, cliquez ici.
정수 김 est un conférencier qui a remporté le prix du meilleur projet de données à la conférence Elastic. Pour plus d'infos, cliquez ici.
Il y a des informations sur les listes et les cartes dans le client Java Elasticsearch. Pour plus d'infos, cliquez ici.
Il y a une session sur les nouvelles fonctionnalités du client Python Elasticsearch. Pour plus d'infos, cliquez ici.
Lisa Jung | 450bb3d9-26e9-4c37-88c2-de5f28cbb93d
Overview Schedule Lisa Jung Developer Advocate | Elastic Lisa Jung is a developer advocate at Elastic. She loves geeking out over data and teaching developers how they can search, analyze, and visualize data with Elastic Stack. Her journey to landing a dream job as a developer advocate has been an u ...
MissingRequiredPropertyException in a response | Elasticsearch Java API Client [7.17] | Elastic | missing-required-property.html
A newer version is available. For the latest information, see thecurrent release documentation.Elastic Docs›Elasticsearch Java API Client [7.17]›Troubleshooting« TroubleshootingNoSuchMethodError RequestOptions$Builder.removeHeader when creating a client »MissingRequiredPropertyException in a respons ...
정수 김 | df68d309-431b-4611-8f8e-e7a09c556021
Overview Schedule 정수 김 상명대학교 2016 ~ 2022 상명대학교 게임전공, 인공지능융합전공 (서울, 종로구)2021 상명 데이터콘서트 최우수상 2021 데이터야놀자 세션 발표2021 Elastic Fundamental Training Talks 대학생의 우당탕탕 엘라스틱 고군분투 이야기
Lists and maps | Elasticsearch Java API Client [7.17] | Elastic | lists-and-maps.html

Lucene text analysis

Index time

Input documents →

Id	Nom
1	Céline
2	Celia

Ascii folding → Celine, Celia

Lowercase → celine, celia

Search time

Nom
CÉLINE

← Search term

CELINE ← Ascii folding

celine ← Lowercase

Index

Key	Document id
celine	1
celia	2

Lucene text analysis

Index time

Input document →

Id	Nom
1	I'm a developer in Berlin
2	She develops software

Lowercase → i'm a developer in berlin
she develops software

Stop → i'm developer berlin
she develops software

Stemmer → i'm develop berlin
she develop softwar

Search time

Nom
Software Development

← Search term

software development ← Lowercase

software development ← Stop

softwar develop ← Stemmer

Index

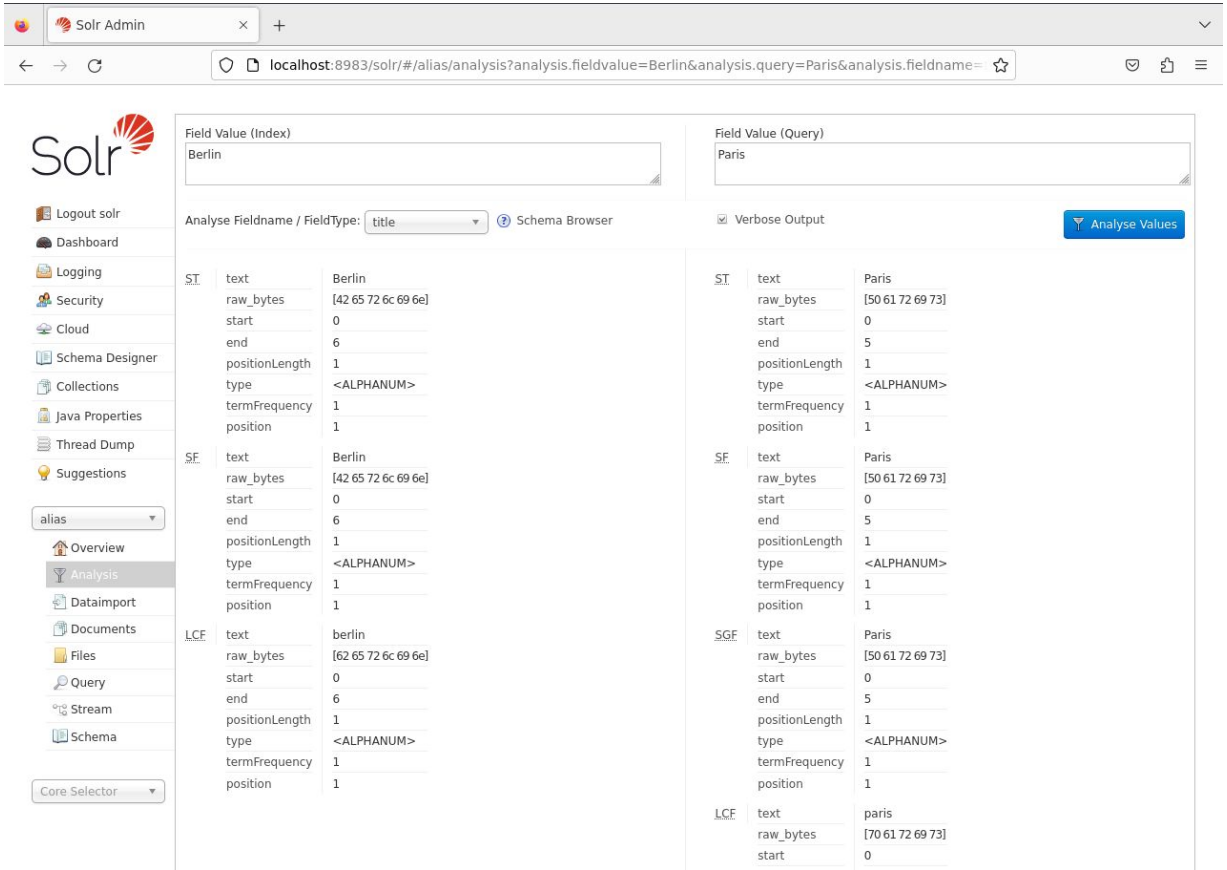
Key	Document id
softwar	2
develop	2
…	…

_analyze API (Elasticsearch and OpenSearch)

```
GET _analyze
{
  "text": [
    "I'm a developer in Berlin"
  ],
  "tokenizer": "standard",
  "filter": [
    "lowercase",
    {
      "type": "stop",
      "stopwords": "_english_"
    },
    {
      "type": "stemmer",
      "language": "english"
    }
  ]
}
```

```
"tokens": [
  {
    "token": "i'm",
    "start_offset": 0,
    "end_offset": 3,
    "type": "<ALPHANUM>",
    "position": 0
  },
  {
    "token": "develop",
    "start_offset": 6,
    "end_offset": 15,
    "type": "<ALPHANUM>",
    "position": 2
  },
  {
    "token": "berlin",
    "start_offset": 19,
    "end_offset": 25,
    "type": "<ALPHANUM>",
    "position": 4
  }
]
```

Analysis menu in the Solr Admin dashboard



Language analyzers - the simple (and wrong) solution

Language analyzer

OpenSearch supports the following language values with the `analyzer` option: `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`, and `thai`.

To use the analyzer when you map an index, specify the value within your query. For example, to map your index with the French language analyzer, specify the `french` value for the analyzer field:

```
"analyzer": "french"
```

https://opensearch.org/docs/latest/analyzers/language-analyzers/

https://www.elastic.co/guide/en/elasticsearch/reference/8.7/analysis-lang-analyzer.html

Stemming

- Lemma : canonical form - dictionary (développeuse → développer)
- Stem : reduced form - algorithmical (développeuse → developeu)

- Stemming is faster and shipped out of the box
- Lemmatization may require a license

- Stemming is largely recommended by Elastic
  - https://www.elastic.co/guide/en/elasticsearch/reference/8.7/stemming.html#dictionary-stemmers
  - https://www.elastic.co/guide/en/elasticsearch/reference/8.7/stemming.html#algorithmic-stemmers

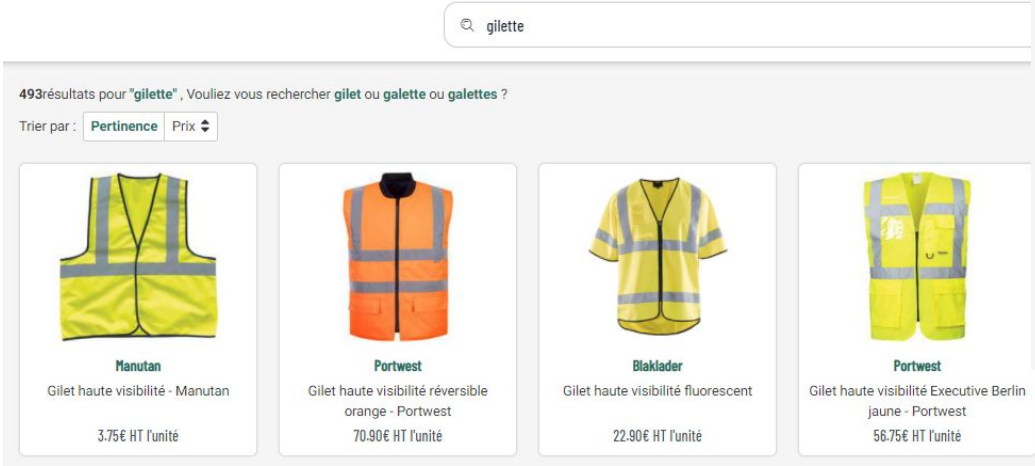
But stemming is not perfect

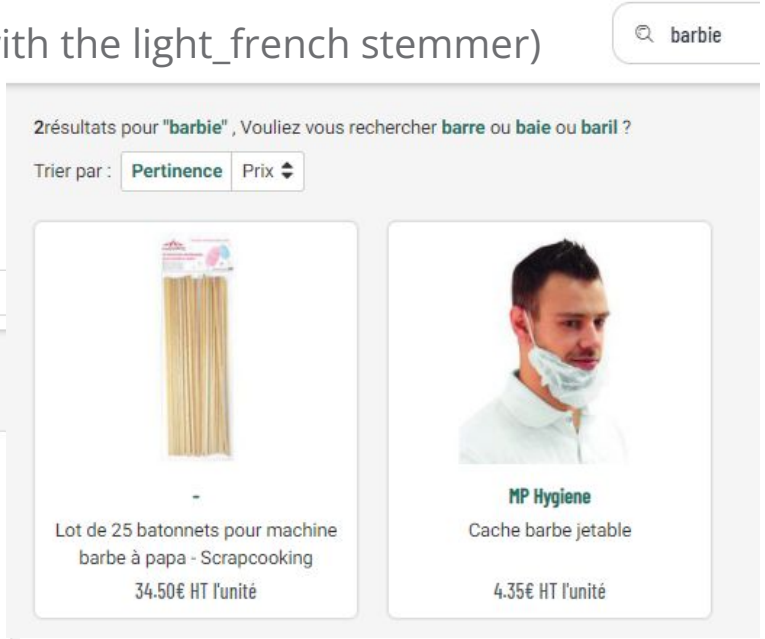
Brands (Gilette = gilet, Barbie = barbe with the light_french stemmer)

Technical limitations (travail = travail, travaux = traval with the light_french stemmer)

Irregular words

Similar spelling (broker = broken)





Details of the convenience language analyzers

```
    "analyzer": {
       "rebuilt german": {
         "tokenizer":  "standard",
         "filter": [
           "lowercase",
           "german_stop",
           "german_keywords",
           "german_normalization",
           "german_stemmer"
         ]
       }
```

```
    "filter": {
       "german_stop": {
         "type":       "stop",
         "stopwords":  "_german_"
       },
       "german_keywords": {
         "type":       "keyword_marker",
         "keywords":   ["Beispiel"]
       },
       "german_stemmer": {
         "type":       "stemmer",
         "language":   "light_german"
       }
    }
```

https://www.elastic.co/guide/en/elasticsearch/reference/8.7/analysis-lang-analyzer.html#german-analyzer

Lucene levers for algorithmic stemmers

Protected words and stemmer override

- Keyword marker
- Stemmer override

And also

- Conditional token filter
- stem_exclusion parameter for language analyzers

Keyword marker

```
GET /_analyze
{
  "tokenizer": "standard",
  "filter": [
    {
      "type": "stop",
      "ignore_case": true,
      "stopwords": [
        "_french_"
      ]
    },
    {
      "type": "stemmer",
      "language": "light_french"
    },
    "lowercase"
  ],
  "text": "Barbie barbe, Gilette gilet, La Croix croissant"
}
```


```
{
  "tokens" : [
    {
      "token" : "barb",
      "start_offset" : 0,
      "end_offset" : 6,
      "type" : "<ALPHANUM>",
      "position" : 0
    },
    {
      "token" : "barb",
      "start_offset" : 7,
      "end_offset" : 12,
      "type" : "<ALPHANUM>",
      "position" : 1
    },
    {
      "token" : "gilet",
      "start_offset" : 14,
      "end_offset" : 21,
      "type" : "<ALPHANUM>",
      "position" : 2
    },
    {
      "token" : "gilet",
      "start_offset" : 22,
      "end_offset" : 27,
      "type" : "<ALPHANUM>",
      "position" : 3
    },
    {
      "token" : "croi",
      "start_offset" : 32,
      "end_offset" : 37,
      "type" : "<ALPHANUM>",
      "position" : 5
    },
    {
      "token" : "croi",
      "start_offset" : 38,
      "end_offset" : 47,
      "type" : "<ALPHANUM>",
      "position" : 6
    }
  ]
}
```

Keyword marker

```
GET /_analyze
{
  "tokenizer": "standard",
  "filter": [
    {
      "type": "stop",
      "ignore_case": true,
      "stopwords": [
        "_french_"
      ]
    },
    {
      "type": "keyword_marker",
      "keywords": [
        "Barbie",
        "Gilette",
        "La Croix"
      ],
      "ignore_case": true
    },
    {
      "type": "stemmer",
      "language": "light_french"
    },
    "lowercase"
  ],
  "text": "Barbie barbe, Gilette gilet, La Croix croissant"
}
```


```
{
  "tokens" : [
    {
      "token" : "barbie",
      "start_offset" : 0,
      "end_offset" : 6,
      "type" : "<ALPHANUM>",
      "position" : 0
    },
    {
      "token" : "barb",
      "start_offset" : 7,
      "end_offset" : 12,
      "type" : "<ALPHANUM>",
      "position" : 1
    },
    {
      "token" : "gilette",
      "start_offset" : 14,
      "end_offset" : 21,
      "type" : "<ALPHANUM>",
      "position" : 2
    },
    {
      "token" : "gilet",
      "start_offset" : 22,
      "end_offset" : 27,
      "type" : "<ALPHANUM>",
      "position" : 3
    },
    {
      "token" : "croi",
      "start_offset" : 32,
      "end_offset" : 37,
      "type" : "<ALPHANUM>",
      "position" : 5
    },
    {
      "token" : "croi",
      "start_offset" : 38,
      "end_offset" : 47,
      "type" : "<ALPHANUM>",
      "position" : 6
    }
  ]
}
```

Stemmer override

```
PUT /my-index-000001
{
  "settings": {
    "analysis": {
      "analyzer": {
        "my_analyzer": {
          "tokenizer": "standard",
          "filter": [ "lowercase", "custom_stems", "porter_stem" ]
        }
      },
      "filter": {
        "custom_stems": {
          "type": "stemmer_override",
          "rules": [
            "running, runs => run",
            "stemmer => stemmer"
          ]
        }
      }
    }
  }
}
```

Dictionary stemmers

- Well suited for
  - Stemming irregular words
  - Discerning between words that are spelled similarly but not related conceptually (broker // broken)
- In practice
  - Algorithmic stemmers typically outperform dictionary stemmers (via Elastic) https://www.elastic.co/guide/en/elasticsearch/reference/8.7/stemming.html#dictionary-stemmers
  - Dictionary quality
  - Size and performance

Lemmatization in e-commerce

- Performs better than stemming if ...
- You have a custom dictionary
- Example for French

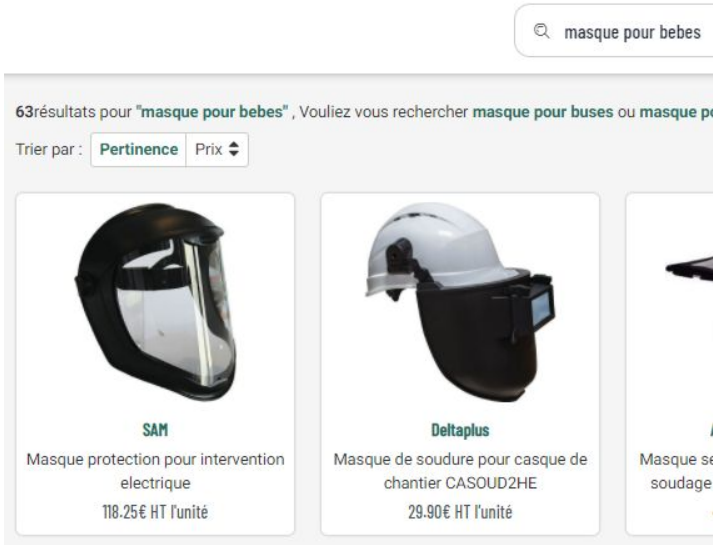
```
"lemmagen_fr" : {
  "type" : "lemmagen",
  "lexicon" : "fr"
}
```

https://github.com/adelean/elasticsearch-analysis-lemmagen

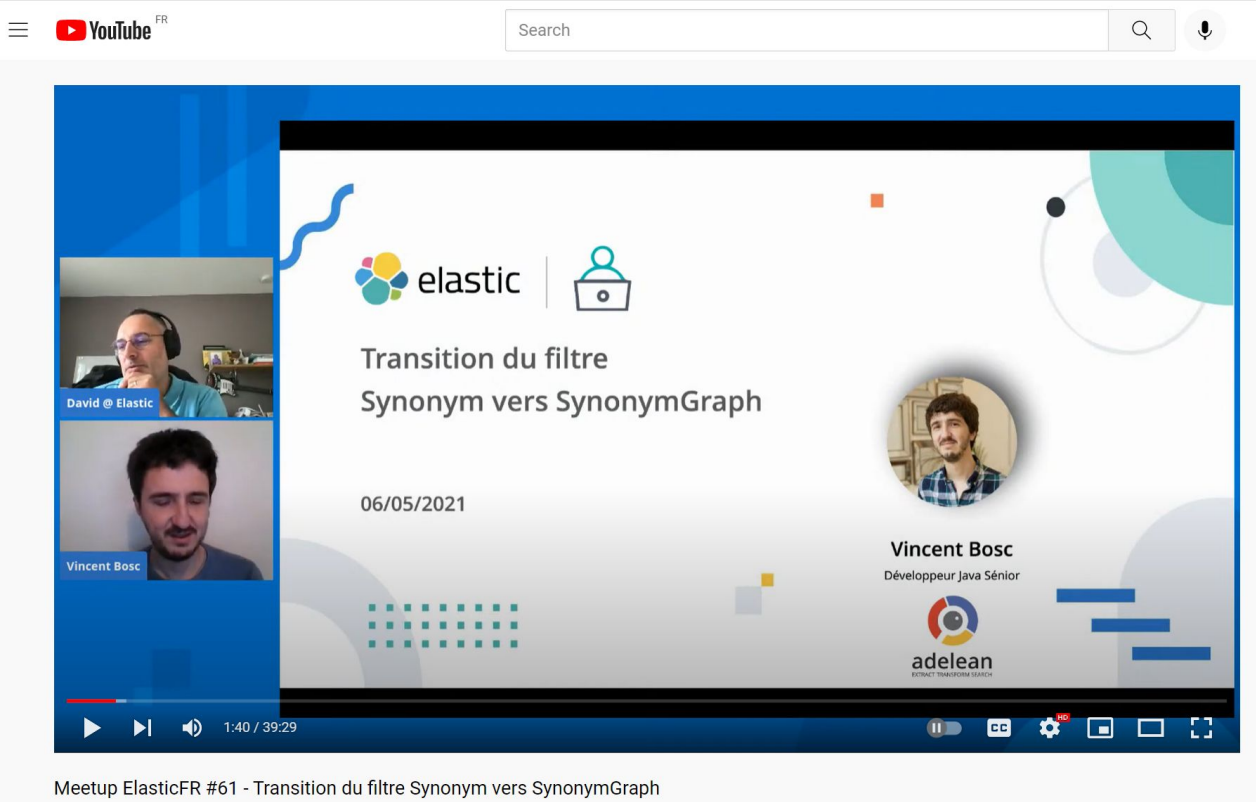
- IMPORTANT! - see License chapter. (https://github.com/hlavki/jlemmagen#markdown-header-license)

Stop words

- Exemples: the, in, a, for, ...
- Side effects when words are not removed :
  - “case for xylophone” should return zero results but might return “case for ...” if a rule like « 2<70% » was used for minimum_should_match
  - another example (French) : « masque pour bébés »
- Issues with the default Elasticsearch implementation
  - Documented here : Meetup ElasticFR #61 - Transition du filtre Synonym vers SynonymGraph - YouTube and here: Bug: When using graph synonym and stop token filter together · Issue #28838 · elastic/elasticsearch · GitHub
  - Workaround : implementing stop words like char_filter



Issues with stopwords and graphs in detail



Meetup ElasticFR #61 - Transition du filtre Synonym vers SynonymGraph

https://www.youtube.com/watch?v=DcjmhkeQt-I

Stop words

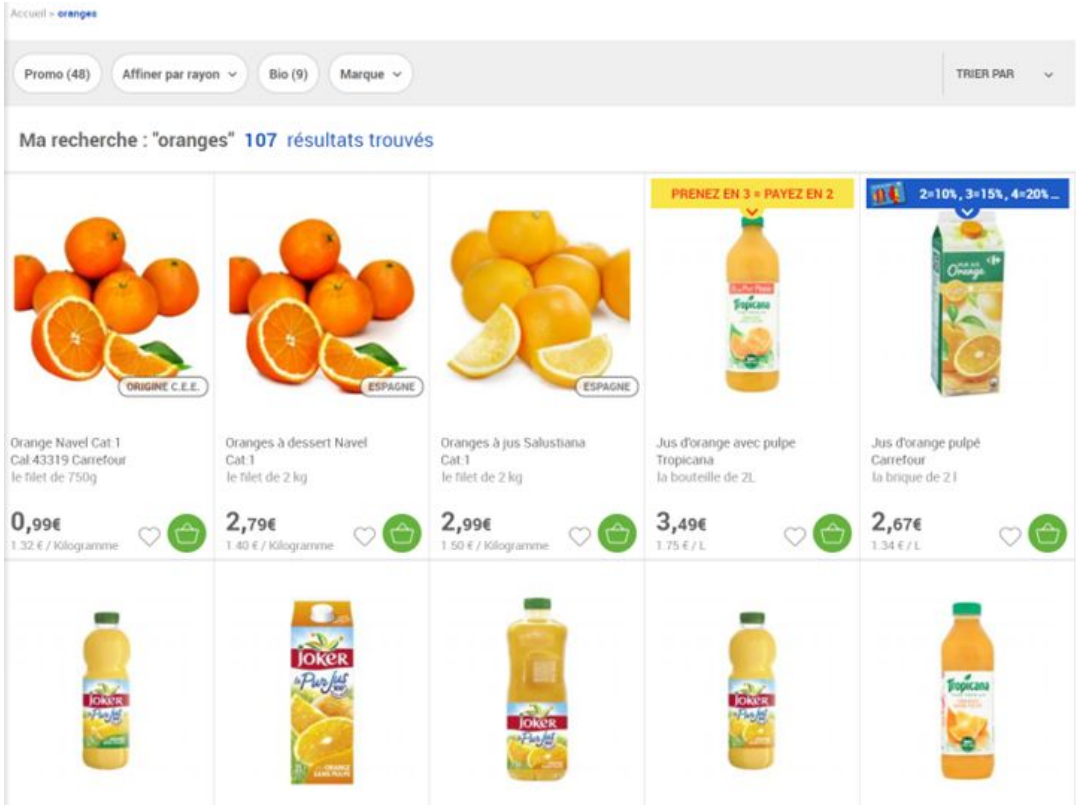
Contournement : implementer les stop words comme char_filter :

```
"mdf_stop_words": {
  "type": "pattern_replace",
  "pattern":
"(\\s|^|\\.|\\-|\\,)(de|la|des|du|pour|le|les|un|une|au|et)(
?=\\s|$|\\.|\\-|\\,)",
  "replacement": " ",
  "flags": "CASE_INSENSITIVE"
}
```

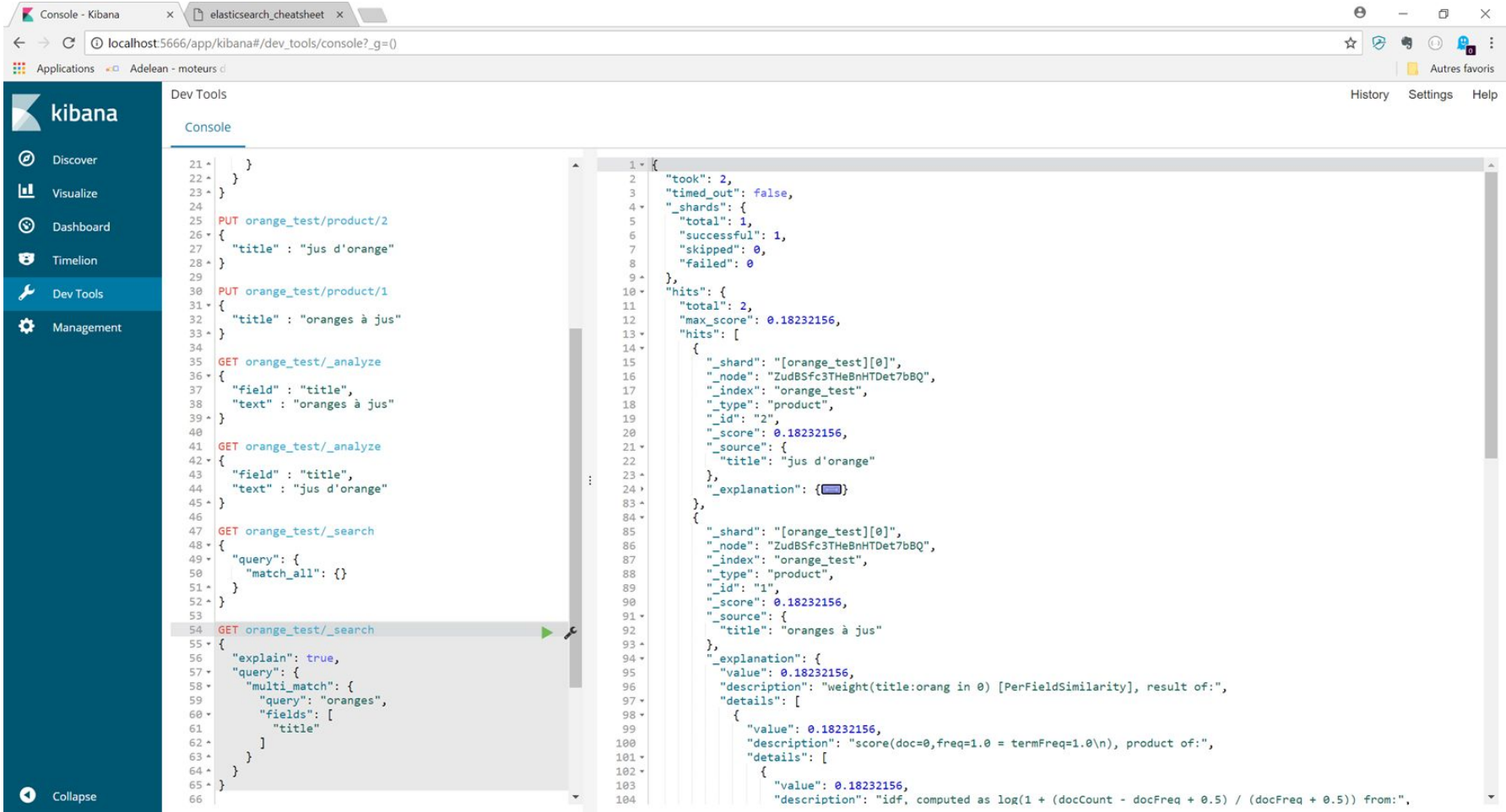
Searching for phrases: detecting the most important word

- orange à jus 🍊 versus jus d'orange 🧃
- The problem:
  - when searching for a single word (ex. oranges) : making sure that the right products pop up first
  - when searching for a phrase and getting zero results : making sure the new search is launched with the most meaningful word
- Simple hack for Latin languages:
  - The most important word is always the first word 🙂

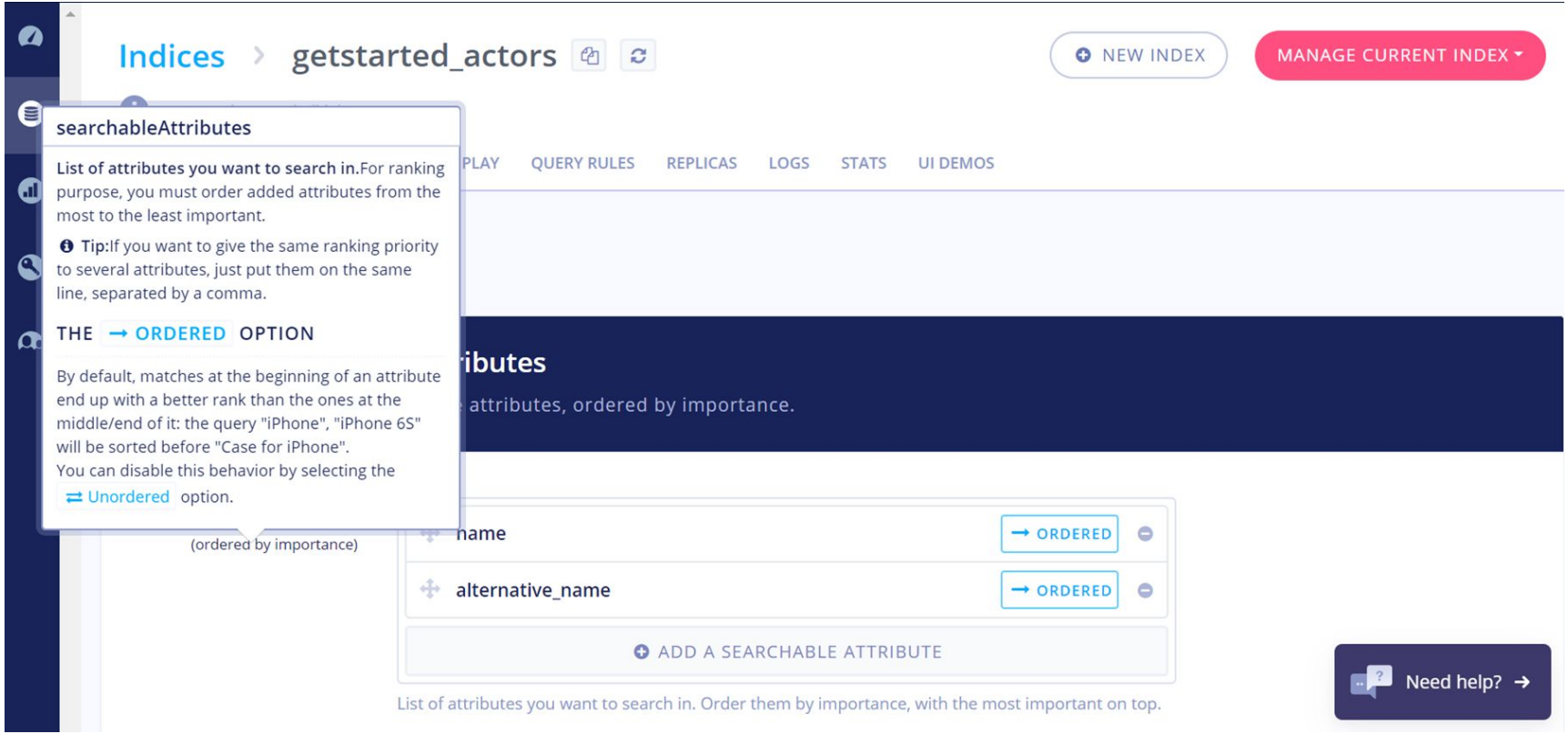
The problem with oranges



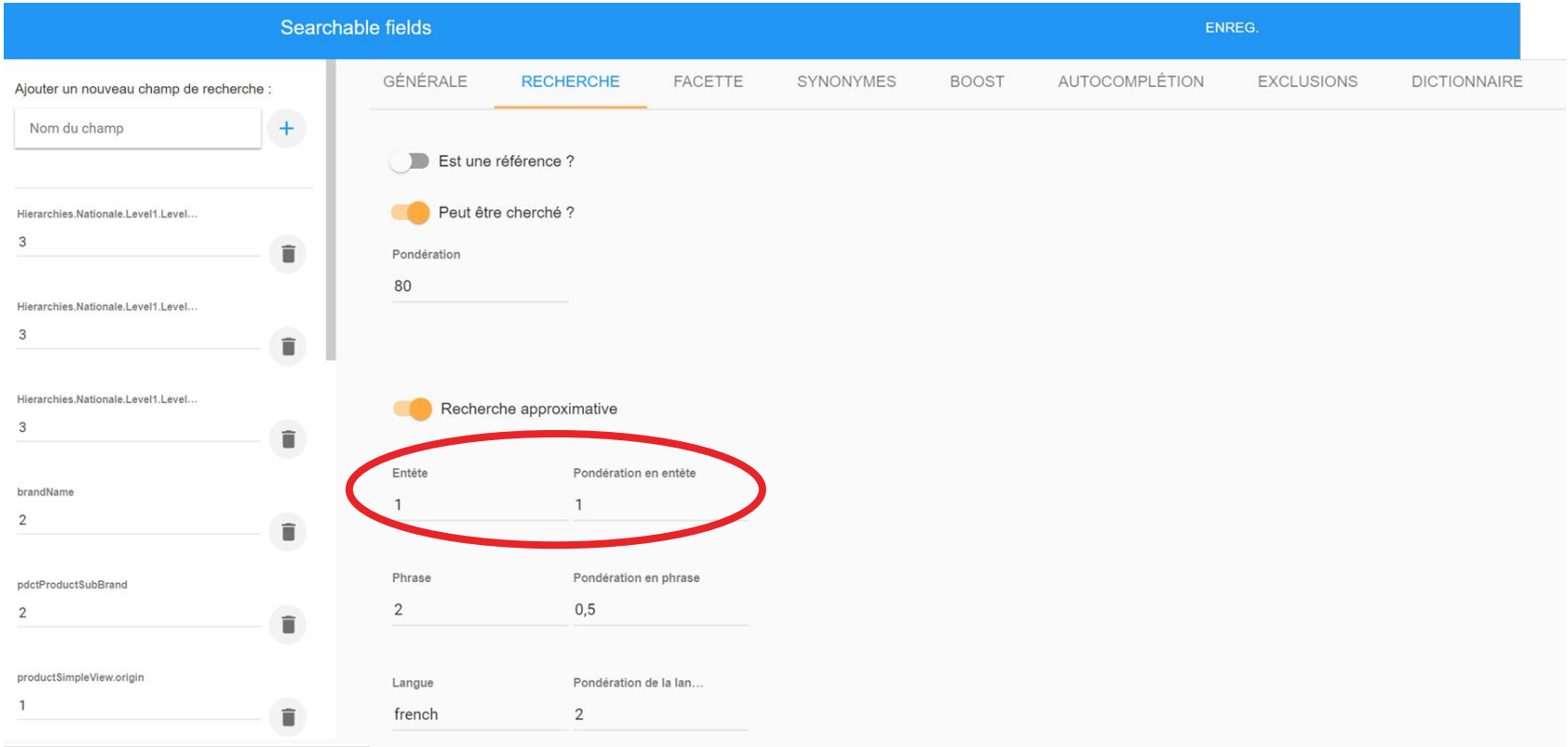
The problem, explained



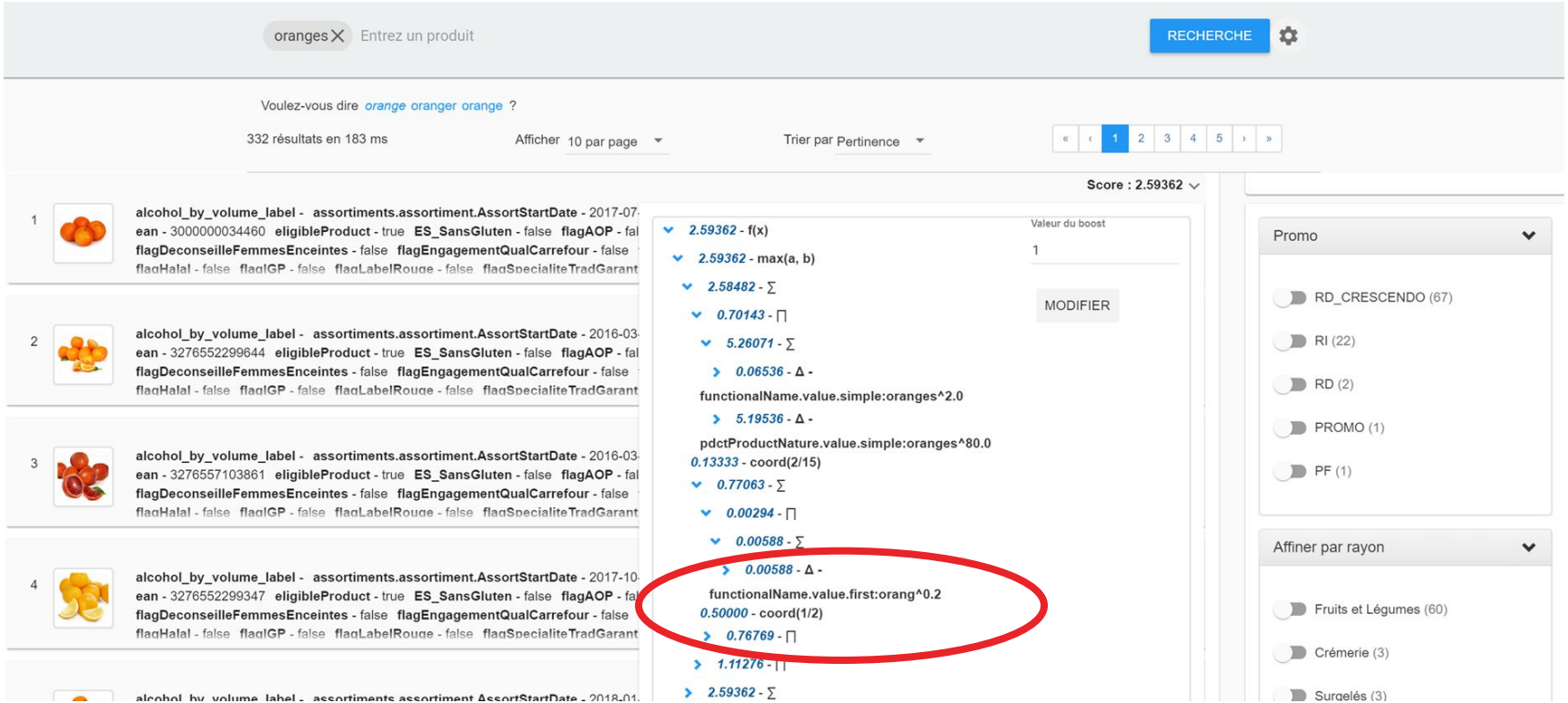
The solution



The solution with Apache Lucene (via Adelean a2)



The solution with Apache Lucene (via Adelean a2)



How about non latin languages?

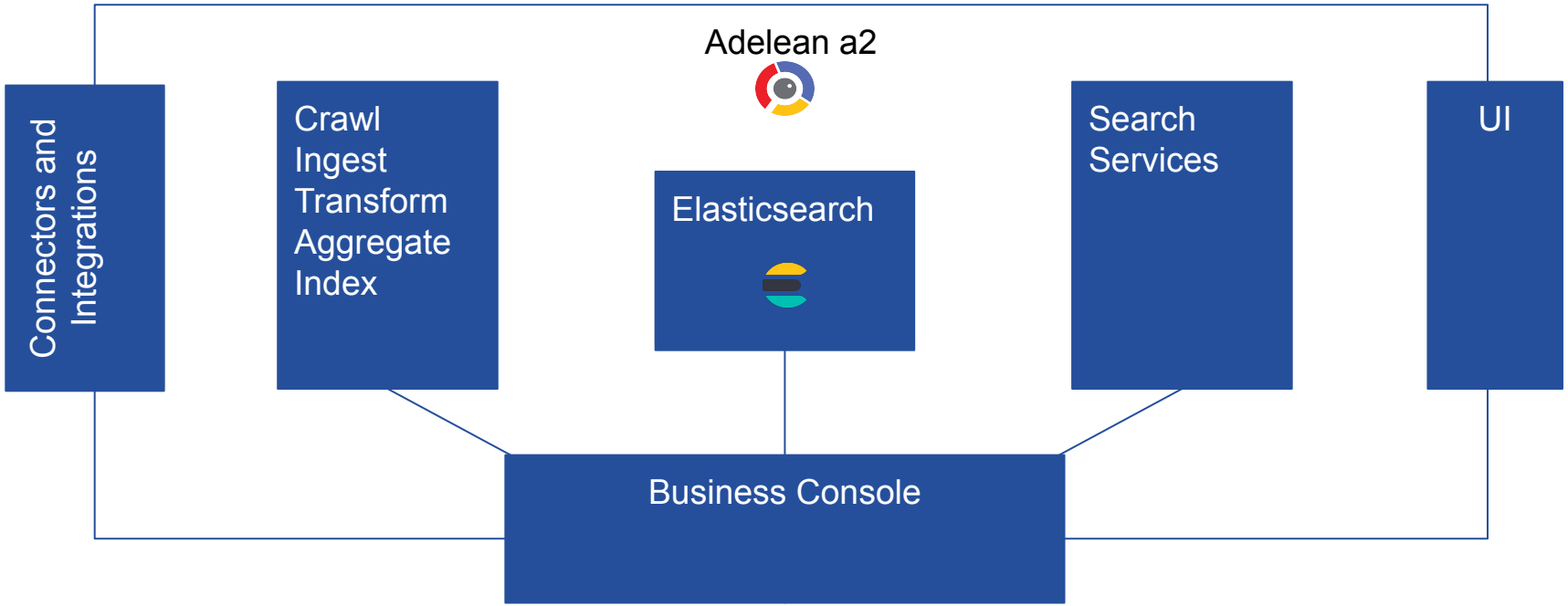
case *for iphone* versus *iphone* ***case***

With our Czech colleagues we produced the following rule (valid for Czech 🇨🇿, English 🇬🇧 and other similar languages):

Place the accent (boost) on the second word UNLESS the phrase contains one of the connecting particles (for, about, on, to, etc. - *specific list to be given in the Business Console for each language*)

boxy *na zavěšení*
boxy - boxes - main subject
na - for - particle
zavešení - hanging - use

The Business Console, window to Lucene config



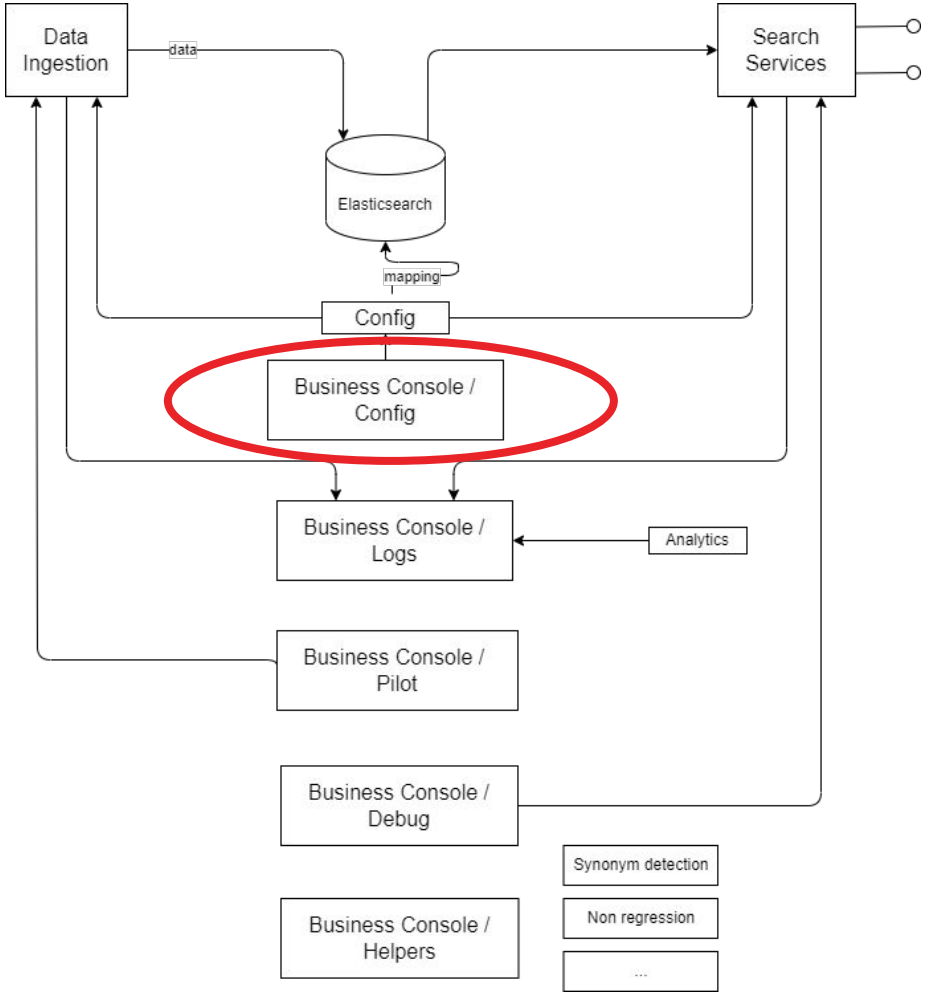
Business Console

Monitoring tools

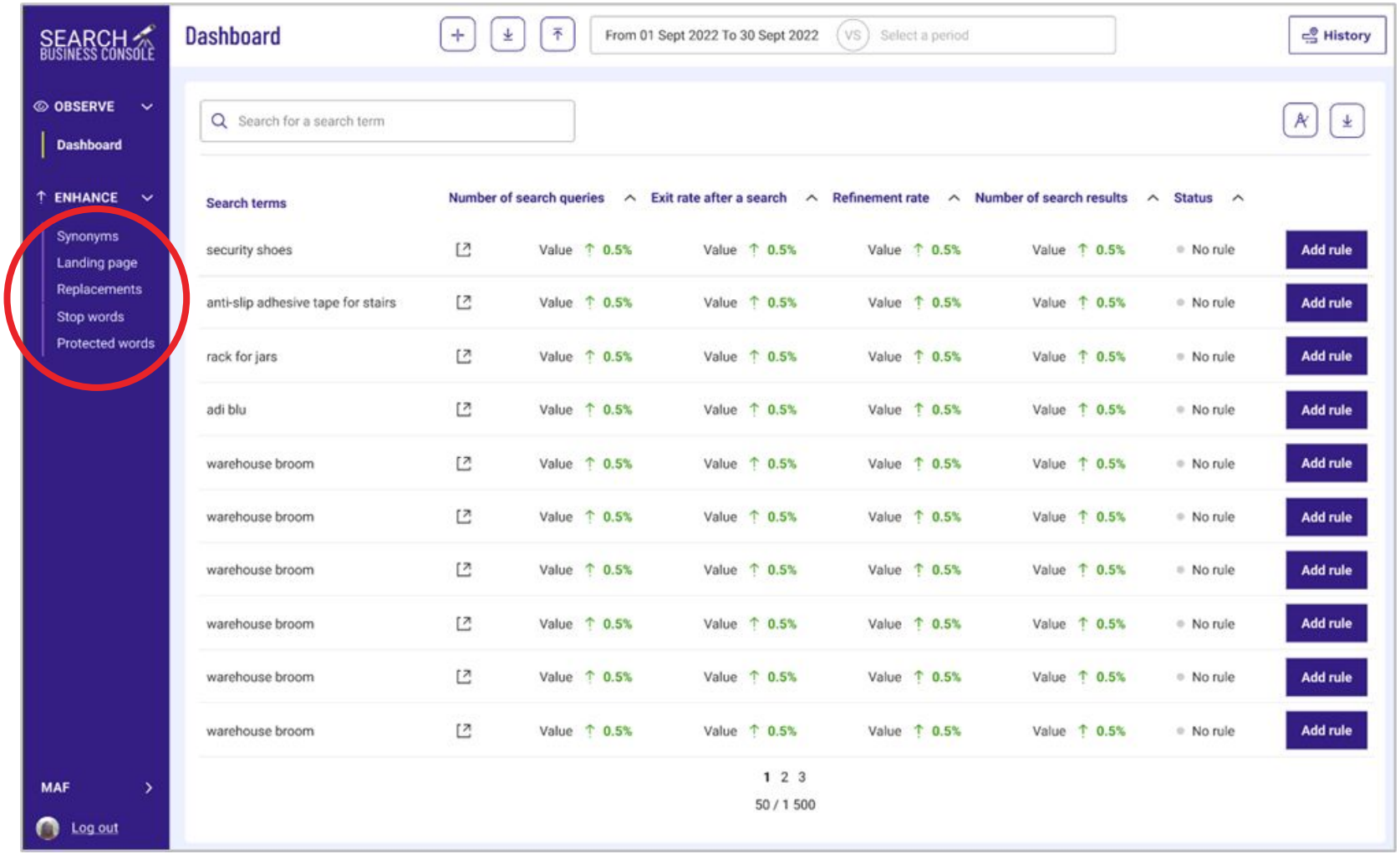
Actions and configurations

Merchandising

...



https://all.site/management/


SEARCH BUSINESS CONSOLE
Dashboard
From 01 Sept 2022 To 30 Sept 2022
VS
Select a period
History
OBSERVE
Dashboard
ENHANCE
Synonyms
Landing page
Replacements
Stop words
Protected words
MAF
Log out
Search for a search term
Search terms
Number of search queries
Exit rate after a search
Refinement rate
Number of search results
Status
security shoes
anti-slip adhesive tape for stairs
rack for jars
adi blu
warehouse broom
Value 0.5%
No rule
Add rule
1 2 3
50 / 1 500

But NLP is better at this task

SIMPLE_SENTENCE = "I want to buy an iPhone case";

SIMPLE_PHRASE_WITH_PARTICLE = "case for iPhone";

SIMPLE_PHRASE = "iPhone case";

COMPLEX_SENTENCE = "I want to buy a small orange Apple iPhone case";

COMPLEX_SENTENCE_YELLOW = "I want to buy a small yellow Apple iPhone case";

Apache OpenNLP

https://github.com/adelean/opennlp-learning

Training data : opennlp-learning/src/main/resources/en-ner-products.train at master · adelean/opennlp-learning (github.com)

Special use cases - Hungarian

```
    "analyzer": {
       "rebuilt hungarian": {
         "tokenizer":  "standard",
         "filter": [
           "lowercase",
           "hungarian stop",
           "hungarian keywords",
           "hungarian_stemmer"
         ]
       }
     }
```

```
  "analysis": {
     "filter": {
       "hungarian_stop": {
         "type":       "stop",
         "stopwords":  "_hungarian_"
       },
       "hungarian keywords": {
         "type":       "keyword_marker",
         "keywords":   ["példa"]
       },
       "hungarian_stemmer": {
         "type":       "stemmer",
         "language":   "hungarian"
       }
     }
```

Hunspell performs better

Special use cases - Hungarian

Hunspell token filter

Provides dictionary stemming based on a provided Hunspell dictionary. The `hunspell` filter requires configuration of one or more language-specific Hunspell dictionaries.

This filter uses Lucene's HunspellStemFilter.

TIP

If available, we recommend trying an algorithmic stemmer for your language before using the `hunspell` token filter. In practice, algorithmic stemmers typically outperform dictionary stemmers. See Dictionary stemmers.

However, hunspell performs better for Hungarian 🙂
Dictionaries (in LibreOffice, LibreOffice extensions, Mozilla Add-Ons)

Special use cases - how about Finnish ?

lentokonesuihkuturbiinimoottoriapumekaanikkoaliupseerioppilas

"airplane jet turbine engine auxiliary mechanic non-commissioned officer student"

https://en.wikipedia.org/wiki/Longest_words

Hunspell does not perform well for Finnish

Voikko Analysis for Elasticsearch
https://github.com/EvidentSolutions/elasticsearch-analysis-voikko deprecated

Raudikko Analysis for Elasticsearch
https://github.com/EvidentSolutions/elasticsearch-analysis-raudikko works with Elasticsearch 8.x

Raudikko Analysis for Elasticsearch

https://github.com/EvidentSolutions/elasticsearch-analysis-raudikko

Parameter	Default value	Description
analyzeAll	true	Use all analysis possibilities or just the first
splitCompoundWords	false	Split analysed compound words to its parts
minimumWordSize	3	minimum length of words to analyze
maximumWordSize	100	maximum length of words to analyze
analysisCacheSize	1024	number of analysis results to cache

Searching for names in a multi-language context

Phonetic analysis plugin https://www.elastic.co/guide/en/elasticsearch/plugins/8.7/analysis-phonetic.html: brings too much noise

- Madonna → MTN
- mouton → MTN

Alternative : IPA or ARPAbet encoding

- https://en.wikipedia.org/wiki/International_Phonetic_Alphabet
- https://en.wikipedia.org/wiki/ARPABET

What is language...

Berlin Buzzwords: bɜl'ɪn b'ʌzwɜdz

Tour Eiffel: tuʁ ɛfɛl

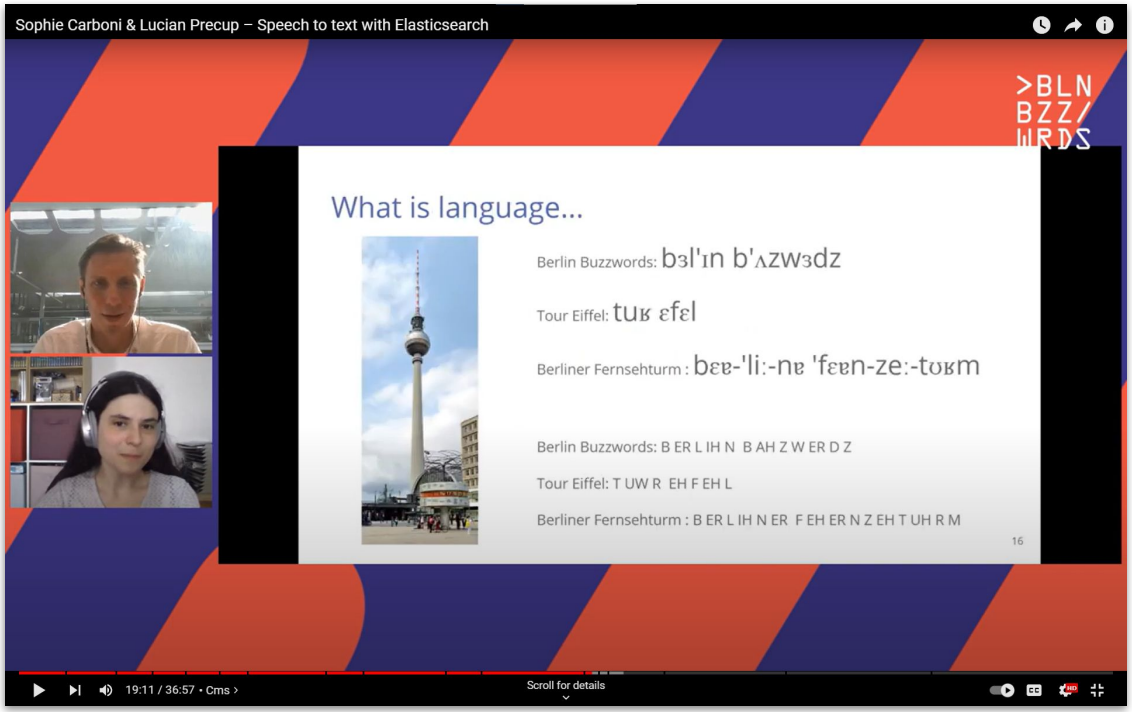
Berliner Fernsehturm : bɛɐ-'liːnɐ 'fɛɐn-zeːtʊʁm

Berlin Buzzwords: B ER L IH N B AH Z W ER D Z

Tour Eiffel: T UW R EH F EH L

Berliner Fernsehturm : B ER L IH N ER F EH ER N Z EH T UH R M

ARPAbet encoding



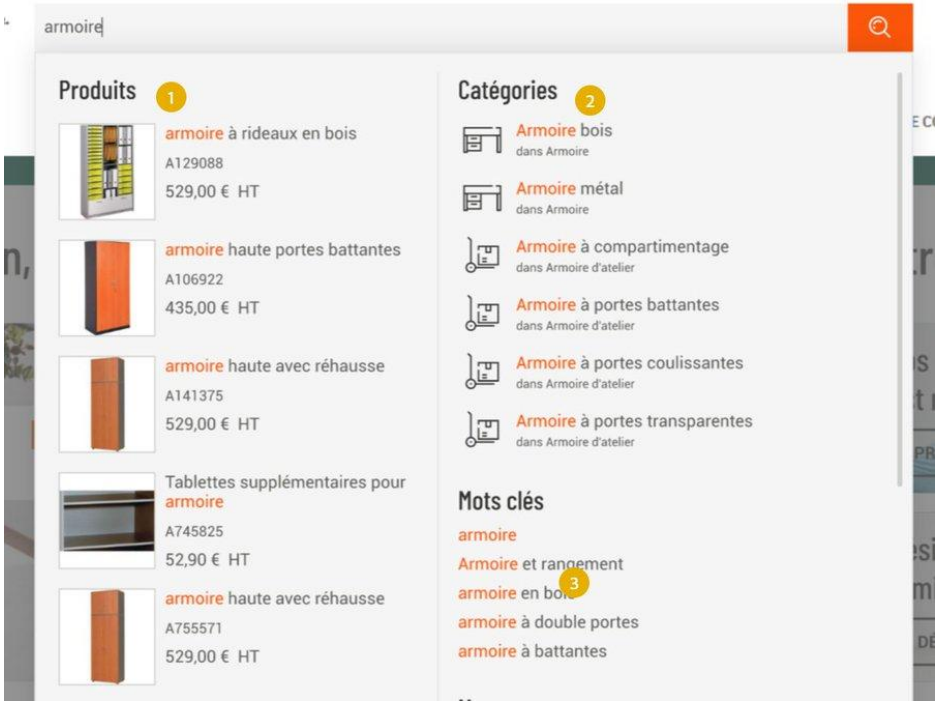
Sophie Carboni & Lucian Precup – Speech to text with Elasticsearch - YouTube

How about compound words?

Solution in two steps:

- 1/ Create a keywords dictionary for auto-completion
- 2/ Also index with ngram
- 3/ Provide a zero results fallback

Step 1 : keywords and key phrases



Solution :
- based on auto-completion #3
- separate index with keywords and keyphrases

Problem : how to generate relevant keywords and keyphrases? Especially when the input data is unstructured.

Auto-completion 3: keywords and key phrases

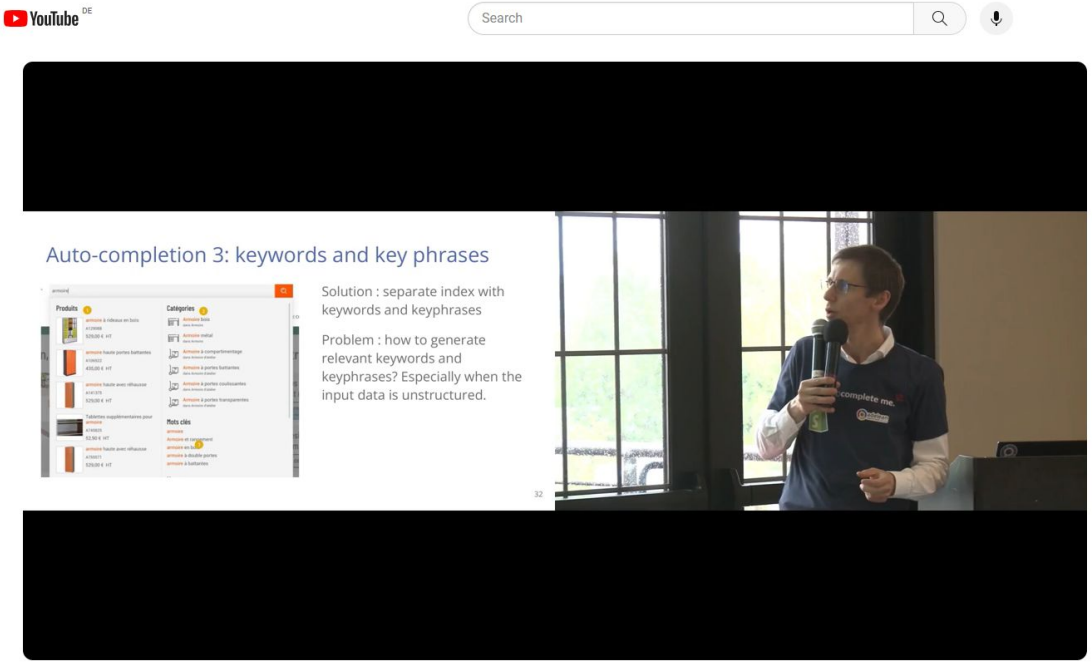
Extract keywords and key phrases from a non-structured text :

- NLP, Machine Learning
- Shingles
- Custom code and usage of analyzers to normalize the data

A community search engine to organize the Internet?

Shingle 2 ⇒ *A* community, community search, search engine, engine *to*, *to* organize, organize *the*, *the* Internet
Shingle 3 ⇒ *A* community search, community search engine, search engine *to*, engine *to* organize, *to* organize *the*, organize the Internet

A story of auto-completions



https://youtu.be/Lo262_tiV9M?t=2114

Step 2: index with ngram

```
"fields" : {
  "ngram" : {
    "type" : "text",
    "store" : true,
    "term_vector" : "with_positions_offsets",
    "analyzer" : "a2_ngram"
  },
  …
```

```
"analysis" : {
  "filter" : {
    "a2_ngram" : {
      "type" : "ngram",
      "min_gram" : "3",
      "max_gram" : "4"
    },
    …
```

Demo in Dev Tools - Elastic (adelean.fr)

Step 3: Provide a zero results fallback

```
POST _a2/search
{
  "catalog": [
    "hyperu"
  ],
  "text": [
    "tomatecerise"
  ]
}
```

Demo in [Dev Tools - Elastic (adelean.fr)]

Thank you for your attention

Questions / Feedback / More …

@lucianprecup

@a2lean

info@adelean.com

http://www.adelean.com

http://www.linkedin.com/company/adelean

http://www.meetup.com/fr-FR/search-and-data