Data Management : Labo 1 - Elasticsearch

Group : D Students : Peiris Sébastien & Raemy Mathis Date : 03.11.2021

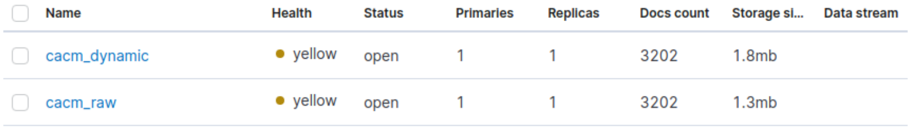
D1 - Creation of an ingest pipeline

```
PUT _ingest/pipeline/Lab1Pipeline
{
  "processors": [
    {
      "csv": {
        "field": "_row",
        "target_fields": [
          "id ",
          "author",
          "title",
          "date",
          "summary"
        ],
        "separator": "\t",
        "quote": "§",
        "empty_value": "null"
      }
    },
    {
      "split": {
        "field": "author",
        "separator": ";",
      }
    },
    {
      "remove": {
        "field": "_row"
      }
    }
  ]
}
```

D2 - Reindex from `cacm_raw` to `cacm_dynamic`

```
POST _reindex
{
  "source": {
  "index": "cacm_raw"
  },
  "dest": {
  "index": "cacm_dynamic",
  "pipeline": "Lab1Pipeline"
  }
}
```

Screenshot of the “Index Management” tab in Kibana showing that there is the same number of document in `cacm_raw` than in `cacm_dynamic` :



□	Name	Health	Status	Primaries	Replicas	Docs count	Storage si...	Data stream
□	cacm_dynamic	● yellow	open	1	1	3202	1.8mb	
□	cacm_raw	● yellow	open	1	1	3202	1.3mb	

D3 - Mapping

Create index. . . :

```
PUT /cacm_standard
{
  "mappings": {
    "properties": {
      "id" : {
        "type": "keyword",
        "index": false,
        "store": true
      },
      "author":{
        "type": "text",
        "fielddata": true,
        "index": true
      },
      "title" : {
        "type": "text",
        "fielddata": true,
        "index": true
      },
      "date" : {
        "type": "date",
        "index": true
      },
      "summary": {
        "type": "text",
        "fielddata": true,
        "index": true,
        "index_options": "offsets"
      }
    }
  }
}
```

. . . and reindex :

```
POST _reindex
{
  "source": {
    "index": "cacm_raw"
  },
  "dest": {
    "index": "cacm_standard",
    "pipeline": "Lab1Pipeline"
  }
}
```

D4 - Term Vector

Creation of the index. . . :

```
PUT /cacm_termvector
{
  "mappings": {
    "properties": {
      "id" : {
        "type": "keyword",
        "index": false,
        "store": true
      },
      "author":{
        "type": "text",
        "fielddata": true,
        "index": true
      },
      "title" : {
        "type": "text",
        "fielddata": true,
        "index": true,
        "term_vector": "with_positions_offsets_payloads"
      },
      "date" : {
        "type": "date",
        "index": true
      },
      "summary": {
        "type": "text",
        "fielddata": true,
        "index": true,
        "index_options": "offsets",
        "term_vector": "with_positions_offsets_payloads"
      }
    }
  }
}
```

. . . and reindex :

```
POST _reindex
{
  "source": {
    "index": "cacm_raw"
  },
  "dest": {
    "index": "cacm_termvector",
    "pipeline": "Lab1Pipeline"
  }
}
```

D5 - Termvector presence check

Command with the id of an article :

```
GET /cacm_termvector/_termvectors/4IXG4HwBncU1UjfZcCq5
{
  "fields" : ["title", "summary"],
  "offsets" : true,
  "payloads" : true,
  "positions" : true,
  "term_statistics" : true,
  "field_statistics" : true
}
```

Returns :

```
{
  "_index" : "cacm_termvector",
  "_type" : "_doc",
  "_id" : "4IXG4HwBncU1UjfZcCq5",
  "_version" : 1,
  "found" : true,
  "took" : 1,
  "term_vectors" : {
      ...
   }
```

```
 }
```

D6 - Question : Term vector

Size of the `cacm_standard` index : 1.6mb Size of the `cacm_termvector` index : 2.5mb

D7 - Question : Discussion about the results of D6

The term vector index contains more information, such as the positions of each term, the start and end character offsets mapping to its origin in the original string or some payloads (cf. elasticsearc doc: term_vector). Since we used the `with_positions_offsets_payloads` parameter, all of the above information is stored in our `cacm_termvector`.

D8 - Author with the highest number of publications

API request :

```
GET /cacm_standard/_search
{
  "size": 0,
  "aggs": {
    "max-pub-author": {
      "terms": {
        "field": "author",
        "order": {
          "_count": "desc"
        }
      }
    }
  }
}
```

This request returns the top 10 of the authors with the most publications in the `cacm_standard` index. Author with the highest number of publications : “j”. Number of publications : 785.

D9 - Top 10 terms in titles

API request :

```
GET /cacm_standard/_search
{
  "size": 0,
  "aggs": {
    "max-pub-title": {
      "terms": {
        "field": "title",
        "order": {
          "_count": "desc"
        }
      }
    }
  }
}
```

Top 10 : 1. of : 1138 2. algorithm : 975 3. a : 895 4. for : 714 5. the : 645 6. and : 434 7. in : 416 8. on : 340 9. an : 275 10. computer : 275

D10 - Creation of the different indices

Without reindex requests.

Index `cacm_whitespace` :

```
PUT /cacm_whitespace
{
  "mappings": {
    "properties": {
      "id" : {
        "type": "keyword",
        "index": false,
        "store": true
      },
      "author":{
        "type": "text",
        "fielddata": true,
        "index": true
      },
      "title" : {
        "type": "text",
        "fielddata": true,
        "index": true,
        "analyzer" : "rebuilt_whitespace"
      },
      "date" : {
        "type": "date",
        "index": true
      },
      "summary": {
        "type": "text",
        "fielddata": true,
        "index": true,
        "index_options": "offsets",
    "analyzer" : "rebuilt_whitespace"
   }
  }
 },
 "settings": {
  "analysis": {
   "analyzer": {
    "rebuilt_whitespace": {
     "tokenizer": "whitespace",
     "filter": [
      ]
    }
   }
  }
 }

}
```

Index cacm_english:

```
PUT /cacm_english
{
 "mappings": {
  "properties": {
   "id" : {
    "type": "keyword",
    "index": false,
    "store": true
   },
   "author":{
    "type": "text",
    "fielddata": true,
    "index": true
   },
   "title" : {
    "type": "text",
    "fielddata": true,
    "index": true,
    "analyzer" : "rebuilt_english"
   },
   "date" : {
    "type": "date",
    "index": true
   },
   "summary": {
    "type": "text",
    "fielddata": true,
        "index": true,
        "index_options": "offsets",
        "analyzer" : "rebuilt_english"
      }
    }
  },
  "settings": {
    "analysis": {
      "filter": {
        "english_stop": {
          "type":       "stop",
          "stopwords":  "_english_"
        },
        "english_keywords": {
          "type":       "keyword_marker",
          "keywords":   ["example"]
        },
        "english_stemmer": {
          "type":       "stemmer",
          "language":   "english"
        },
        "english_possessive_stemmer": {
          "type":       "stemmer",
          "language":   "possessive_english"
        }
      },
      "analyzer": {
        "rebuilt_english": {
          "tokenizer":  "standard",
          "filter": [
            "english_possessive_stemmer",
            "lowercase",
            "english_stop",
            "english_keywords",
            "english_stemmer"
          ]
        }
      }
    }
  }

}
```

Index `cacm_custom_standard_shingles_1-2` :

```
PUT /cacm_custom_standard_shingles_1-2
{
```

```
"mappings": {
  "properties": {
    "id" : {
      "type": "keyword",
      "index": false,
      "store": true
    },
    "author":{
      "type": "text",
      "fielddata": true,
      "index": true
    },
    "title" : {
      "type": "text",
      "fielddata": true,
      "index": true,
      "analyzer" : "custom_standard_1-2_shingles_analyzer"
    },
    "date" : {
      "type": "date",
      "index": true
    },
    "summary": {
      "type": "text",
      "fielddata": true,
      "index": true,
      "index_options": "offsets",
      "analyzer" : "custom_standard_1-2_shingles_analyzer"
    }
  }
},
"settings": {
  "analysis": {
    "analyzer": {
      "custom_standard_1-2_shingles_analyzer": {
        "tokenizer": "standard",
        "filter": [
          "shingle_filter_1-2"
          ]
      }
    },
    "filter": {
      "shingle_filter_1-2": {
        "type": "ngram",
        "min_gram": 1,
        "max_gram": 2
        }
      }
    }
  }

}
```

Index `cacm_custom_standard_shingles_3` :

```
PUT /cacm_custom_standard_shingles_3
{
  "mappings": {
    "properties": {
      "id" : {
        "type": "keyword",
        "index": false,
        "store": true
      },
      "author":{
        "type": "text",
        "fielddata": true,
        "index": true
      },
      "title" : {
        "type": "text",
        "fielddata": true,
        "index": true,
        "analyzer" : "custom_standard_3_shingles_analyzer"
      },
      "date" : {
        "type": "date",
        "index": true
      },
      "summary": {
        "type": "text",
        "fielddata": true,
        "index": true,
        "index_options": "offsets",
        "analyzer" : "custom_standard_3_shingles_analyzer"
      }
    }
  },
  "settings": {
    "analysis": {
      "analyzer": {
        "custom_standard_3_shingles_analyzer": {
          "tokenizer": "standard",
          "filter": [
            "shingle_filter_3"
            ]
        }
      },
      "filter": {
        "shingle_filter_3": {
          "type": "ngram",
          "min_gram": 3,
          "max_gram": 3
        }
      }
    }
  }

}
```

Index `cacm_stop` :

```
PUT /cacm_stop
{
  "mappings": {
    "properties": {
      "id" : {
        "type": "keyword",
        "index": false,
        "store": true
      },
      "author":{
        "type": "text",
        "fielddata": true,
        "index": true
      },
      "title" : {
        "type": "text",
        "fielddata": true,
        "index": true,
        "analyzer" : "stop_analyzer"
      },
      "date" : {
        "type": "date",
        "index": true
      },
      "summary": {
        "type": "text",
        "fielddata": true,
        "index": true,
        "index_options": "offsets",
        "analyzer" : "stop_analyzer"
      }
    }
  },
  "settings": {
    "analysis": {
      "analyzer": {
        "stop_analyzer": {
          "type": "stop",
          "stopwords_path": "data/common_words.txt"
        }
      }
    }
  }

}
```

D11 - Differences between the analyzers

Standard : Divides text into terms according to the Unicode Text Segmentation algorithm.

Whitespace : Divides text into terms when there is a whitespace character.

English : The english analyzer, as well as all other languages analyzers implements a `stem_exclusion` parameter which allows to specify an array of lowercase words that should not be stemmed. This list is specific to the english language.

Custom with 1-2 shingles output : the `ngram` token filter to convert text into 1 or 2 character n-grams.

Custom with 3 shingles output : the `ngram` token filter to convert text into 3 character n-grams.

Stop : Divides text into terms when there is a character that is not a letter and removes a list of stop words.

D12 - Statistics

`cacm_whitespace` : a) Number of indexed documents : 3202 b) Number of indexed terms in the summary field : 90426 c) Top 10 frequent terms of the summary field in the index (in order) : `[of, the, is, and, a, to, in, for, The, are]` d) Size of the index on disk : 1798059 bytes e) Required time for indexing : 541ms

`cacm_english` : a) Number of indexed documents : 3202 b) Number of indexed terms in the summary field : 66582 c) Top 10 frequent terms of the summary field in the index (in order) : `[which, us, comput, program, system, present,` describ, paper, can, gener] d) Size of the index on disk : 1489677 bytes e) Required time for indexing : 611ms

cacm_custom_standard_shingles_1-2 : a) Number of indexed documents : 3202 b) Number of indexed terms in the summary field : 251112 c) Top 10 frequent terms of the summary field in the index (in order) : [e, n, o, a, d, i, m, r, s, t] d) Size of the index on disk : 4831630 bytes e) Required time for indexing : 731ms

cacm_custom_standard_shingles_3 : a) Number of indexed documents : 3202 b) Number of indexed terms in the summary field : 302318 c) Top 10 frequent terms of the summary field in the index (in order) : [the, ion, tio, ing, ati, and, ent, for, pro, ons] d) Size of the index on disk : 3159900 e) Required time for indexing : 722ms

cacm_stop : a) Number of indexed documents : 3202 b) Number of indexed terms in the summary field : 56386 c) Top 10 frequent terms of the summary field in the index (in order) : [computer, system, paper, presented, time, program, data, method, algorithm, discussed] d) Size of the index on disk : 1471078 bytes e) Required time for indexing : 301ms

D13 - Conclusions

1) When singles are too short, the terms split is done letter by letter. It increase the size of the index and a lot of terms repetitions.
2) With the index cacm_whitespace we can see that the terms "the" and the term "The" is considered as a different index because as specified in documentation, it does not lowercase terms,
3) We can see that on analyzer that accept every words that the mosts frequents terms are not really significativ. Those words are probably the same for every english texts. Analyzers that denie the most common words of english are give a better idea of the subject of thoses documents.

D14 - Query string query

1. Publications with the term "Information Retrieval"

```
GET /cacm_english/_search
{
  "query": {
    "query_string": {
      "query": "Information Retrieval"
    }
  }
}
```

2. Publications with both "Information" and "Retrieval"

```
GET /cacm_english/_search
```

```
{
  "query": {
    "query_string": {
      "query": "Information AND Retrieval"
    }
  }
}
```

3. Publications with at least the term “Retrieval” and possibly “Information” but not “Database”

```
GET /cacm_english/_search
{
  "query": {
    "query_string": {
      "query": "Retrieval AND NOT Database"
    }
  }
}
```

4. Publications that starts with “Info”

```
GET cacm_english/_search
{
  "query": {
    "prefix": {
      "summary": {
        "value": "Info"
      }
    }
  }
}
```

5. Publications containing the term “Information” close to “Retrieval”

```
GET /cacm_english/_search
{
  "query": {
    "query_string": {
      "query": "Information Retrieval ~5"
    }
  }
}
```

D15 - Total number of results

1. 287 hits
2. 48 hits
3. 86 hits

4. 0 hits
5. 287 hits

D16 - Index with custom scoring

```
PUT /cacm_similarities
{
  "mappings": {
    "properties": {
      "id" : {
        "type": "keyword",
        "index": false,
        "store": true
      },
      "author":{
        "type": "text",
        "fielddata": true,
        "index": true
      },
      "title" : {
        "type": "text",
        "fielddata": true,
        "index": true
      },
      "date" : {
        "type": "date",
        "index": true
      },
      "summary": {
        "type": "text",
        "fielddata": true,
        "index": true,
        "index_options": "offsets",
        "similarity": "scripted_tfidf"
      }
    }
  },
   "settings": {
      "number_of_shards": 1,
      "similarity": {
        "scripted_tfidf": {
          "type": "scripted",
          "script": {
            "source": "double tf = 1+Math.log(doc.freq); double idf = Math.log((field.docCou
          }
        }
      }
    }
  }
```

D17-18 - Top 10 results & scores with and without custom scoring and API requests

Without custom scoring API request :

```
GET /cacm_standard/_search/
{
  "from" : 0,
  "size" : 10,
  "query": {
  "query_string": {
    "fields": ["summary"],
    "query": "compiler program"
    }
  },
  "fields": [
    "score",
    "title"
  ],
  "_source": false
}
```

Top 10 : 1. “Compilation for Two Computers with NELIAC” : 7.1 2. “Optimizing Bit-time Computer Simulation” : 7.01 3. “An Algebraic Compiler for the FORTRAN Assembly Program” : 6.96 4. “Program Translation Viewed as a General Data Processing Problem” : 6.93 5. “Regular Expression Search Algorithm” : 6.74 6. “WATFOR-The University of Waterloo FORTRAN IV Compiler” : 6.49 7. “Design and Implementation of a Diagnostic Compiler for PL/I” : 6.44 8. “A Parser-Generating System for Constructing Compressed Compilers” : 6.13 9. “The COBOL Librarian - A Key to Object Program Efficiency” : 6.07 10. “Some Techniques Used in the ALCOR ILLINOIS 7090” : 5.91

With custom scoring API request :

```
GET /cacm_similarities/_search/
{
  "from" : 0,
  "size" : 10,
  "query": {
  "query_string": {
    "fields": ["summary"],
    "query": "compiler program"
    }
  },
  "fields": [
    "score",
    "title"
  ],
  "_source": false
}
```

Top 10 : 1. “Design and Implementation of a Diagnostic Compiler for PL/I” : 14.18 2. “WATFOR-The University of Waterloo FORTRAN IV Compiler” : 12.43 3. “Regular Expression Search Algorithm” : 11.3 4. “A Parser-Generating System for Constructing Compressed Compilers” : 11.3 5. “Compilation for Two Computers with NELIAC” : 11.23 6. “Optimizing Bit-time Computer Simulation” : 11.23 7. “Program Translation Viewed as a General Data Processing Problem” : 11.23 8. “Some Techniques Used in the ALCOR ILLINOIS 7090” : 9.9 9. “A Case Study of a New Code Generation Technique for Compilers” : 9.9 10. “A Microprogrammed Implementation of EULER on IBM System/360 Model 30” : 9.55