MAC - Labo3: Indexing and Search with Elasticsearch

2 Indexing and Searching the CACM collection

2.2 Indexing

D.1

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

POST _reindex
{
  "source": {
    "index": "cacm_dynamic"
  },
  "dest": {
    "index": "cacm_standard"
  }
}
```

D.2

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

POST _reindex
{
  "source": {
    "index": "cacm_dynamic"
  },
  "dest": {
    "index": "cacm_termvector"
  }
}
```

D.3

```
GET /cacm_termvector/_termvectors/KQXq1X8BmNg66-2tWovg?fields=summary
```

D.4

Un `term vector` contient des informations et statistiques à propos du champ sur lequel il a été appliqué (`summary` dans notre cas). Il compte notamment le nombre de termes, leurs positions, leurs fréquences etc. Il est possible de modifier les informations reçues en changeant le type de la clé `term_vector` lors de la création de l'index.

D.5

Taille de l'index `cacm_standard` : **1.66mb**

Taille de l'index `cacm_termvector` : **2.39mb**

L'index avec un term_vector `with_offsets` sur le champ *summary* est environ **44%** plus lourd (0.73mb) que le standard. Ceci est facilement compréhensible quand nous observons le nombre d'informations supplémentaires obtenues. En choisissant d'autres valeurs comme `with_position_offsets`, on rajoute des informations pour chaque terme et donc on augmente encore plus la taille de l'index.

Cependant, pour notre collection, la taille n'a pas subi une très grande augmentation car de nombreux documents n'ont tout simplement pas de champ `summary`.

2.3 Reading Index

D.6

> Who is the author with the highest number of publications? How many publications does he/she have?

```
GET cacm_standard/_search
{
  "size": 0,
  "aggs": {
    "authors": {
      "terms": {
        "field": "author",
        "size": 1
      }
    }
  }
}
```

Il s'agit de **Thacher Jr., H. C.** avec **38** publications.

D.7

> List the top 10 terms in the title field together with their frequency.

```
GET cacm_standard/_search
{
  "size": 0,
  "aggs": {
    "titles": {
      "terms": {
        "field": "title",
        "size": 10
      }
    }
  }
}
```

Voici les 10 termes avec leur fréquence:

#	Titre	Fréquence
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

2.4 Using different Analyzers

D.8

Whitespace analyzer

```
PUT cacm_whitespace
{
  "settings": {
    "analysis": {
      "analyzer": {
        "default": {
          "type": "whitespace"
        }
      }
    }
  },
  "mappings": {
      "properties": {
        "id": {
          "type": "keyword",
```

```
          "index": false,
          "store": true
        },
        "author": {
          "type": "keyword"
        },
        "title": {
          "type": "text",
          "fielddata": true
        },
        "date": {
          "type": "date"
        },
        "summary": {
          "type": "text",
          "fielddata": true,
          "index_options": "offsets"
        }
      }
    }
}

POST _reindex
{
  "source": {
    "index": "cacm_dynamic"
  },
  "dest": {
    "index": "cacm_whitespace"
  }
}
```

English analyzer

```
PUT cacm_english
{
  "settings": {
    "analysis": {
      "analyzer": {
        "default": {
          "type": "english"
        }
      }
    }
  },
  "mappings": {
      "properties": {
        "id": {
          "type": "keyword",
          "index": false,
          "store": true
        },
        "author": {
          "type": "keyword"
        },
        "title": {
          "type": "text",
          "fielddata": true
        },
        "date": {
          "type": "date"
        },
        "summary": {
          "type": "text",
          "fielddata": true,
          "index_options": "offsets"
        }
      }
    }
}

POST _reindex
{
  "source": {
    "index": "cacm_dynamic"
  },
  "dest": {
    "index": "cacm_english"
  }
}
```

Custom analyzer with lowercase unigrams and shingles of size 2

```
PUT cacm_custom1
{
  "settings": {
    "analysis": {
      "analyzer": {
        "default": {
          "type": "custom",
          "tokenizer": "standard",
          "filter": [
            "lowercase",
            "shingle_filter"
          ]
        }
      },
      "filter": {
        "shingle_filter": {
          "type": "shingle",
          "max_shingle_size": 2,
          "output_unigrams": true
        }
      }
    }
  },
  "mappings": {
    "properties": {
      "id": {
        "type": "keyword",
```

```
        "index": false,
        "store": true
      },
      "author": {
        "type": "keyword"
      },
      "title": {
        "type": "text",
        "fielddata": true
      },
      "date": {
        "type": "date"
      },
      "summary": {
        "type": "text",
        "fielddata": true,
        "index_options": "offsets"
      }
    }
  }
}

POST _reindex
{
  "source": {
    "index": "cacm_dynamic"
  },
  "dest": {
    "index": "cacm_custom1"
  }
}
```

Custom analyzer with shingles of size 3

```
PUT cacm_custom2
{
  "settings": {
    "analysis": {
      "analyzer": {
        "default": {
          "type": "custom",
          "tokenizer": "standard",
          "filter": [
            "shingle_filter"
          ]
        }
      },
      "filter": {
        "shingle_filter": {
          "type": "shingle",
          "max_shingle_size": 3,
          "min_shingle_size": 3,
          "output_unigrams": false
        }
      }
    }
  },
  "mappings": {
    "properties": {
      "id": {
        "type": "keyword",
        "index": false,
        "store": true
      },
      "author": {
        "type": "keyword"
      },
      "title": {
        "type": "text",
        "fielddata": true
      },
      "date": {
        "type": "date"
      },
      "summary": {
        "type": "text",
        "fielddata": true,
        "index_options": "offsets"
      }
    }
  }
}

POST _reindex
{
  "source": {
    "index": "cacm_dynamic"
  },
  "dest": {
    "index": "cacm_custom2"
  }
}
```

Custom stop analyzer with list of stopwords from `common_words.txt`

```
PUT cacm_stop
{
  "settings": {
    "analysis": {
      "analyzer": {
        "default": {
          "type": "custom",
          "tokenizer": "standard",
          "filter": [
            "stop_filter"
          ]
        }
      },
      "filter": {
        "stop_filter": {
```

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

POST _reindex
{
  "source": {
    "index": "cacm_dynamic"
  },
  "dest": {
    "index": "cacm_stop"
  }
}
```

D.9

Whitespace:

Coupe le texte en terme à chaque fois qu'il rencontre un caractère d'espace.

English:

Utilise une liste de stopwords compris notamment de bouts de mots spécifiques à la langue anglaise afin de regrouper plusieurs mots.

Ex: "gener" qui est le préfixe de nombreux mots (generate, generation, generated...)

Custom-1:

Utilise le standard tokenizer et analyzer qui divisent le texte en termes en utilisant l'algorithme *Unicode Text Segmentation*. L'analyzer Custom-1 contient deux filtres, un qui passe les tokens en minuscule et un autre qui concatène les tokens en termes de taille de 2 au maximum et qui accepte de faire des termes composés d'un seul mot.

Custom-2:

Est similaire au Custom-1. Les seules différences se trouvent dans la suppression du filtre spécifique du lowercase et sur la taille des shingles qui doit être exactement de 3.

Custom-stop:

Utilise le standard tokenizer et un filtre prenant une liste de stopwords customisée de certains mots clés communs de la langue anglaise ainsi que toutes les lettres de l'alphabet.

D.10

Analyzer	whitespace	english	custom-1	custom-2	custom-stop
Nombre de documents indexés	3202	3202	3202	3202	3202
Nombre de termes indexés dans le champ summary	103'275	72'298	237'189	144'719	66'555
Top 10 des termes les plus fréquents du champ summary dans l'index.	1. of (1534) 2. the (1501) 3. is (1382) 4. and (1369) 5. a (1321) 6. to (1293) 7. in (1188) 8. for (1167) 9. The (1072) 10. are (1022)	1. which (781) 2. us (778) 3. comput (663) 4. program (635) 5. system (586) 6. present (514) 7. describ (505) 8. paper (428) 9. can (421) 10. gener (411)	1. the (1541) 2. of (1534) 3. a (1426) 4. is (1384) 5. and (1376) 6. to (1301) 7. in (1234) 8. for (1182) 9. are (1025) 10. of the (938)	1. the number of (97) 2. the use of (86) 3. in terms of (82) 4. a set of (74) 5. is shown that (71) 6. It is shown (64) 7. In this paper (63) 8. as well as (63) 9. This paper describes (55) 10. shown to be (53)	1. The (1072) 2. A (690) 3. This (465) 4. computer (441) 5. system (429) 6. paper (421) 7. presented (372) 8. time (354) 9. program (339) 10. data (309)
Taille de l'index sur le disque	1.79 mb	1.5 mb	3.54 mb	3.77 mb	1.52 mb
Temps requis pour l'indexation	320 ms	440 ms	686 ms	463 ms	326 ms

D.11

La taille sur le disque est liée au nombre de termes indexés et à leur taille. Nous pouvons dire que le temps d'indexation est plutôt légèrement influencé par la taille de l'index si nous prenons en compte une certaine marge d'erreur pour la mesure et le type de machine utilisée.

Nous observons aussi que les analyzers avec des listes de termes en contiennent beaucoup moins que les autres. C'est plutôt logique car les indexeurs ne travaillent qu'avec un ensemble réduit de termes, limitant dès le départ leur nombre. Les analyzers avec les shingles contiennent le plus de termes. Là aussi, c'est un résultat prévisible car, ici, nous prenons chaque variation des n-grammes pour l'indexation. Par exemple, pour *custom-1,* nous aurions aussi deux unigrammes pour un shingle d'une taille de deux. Concrètement, en plus d'indexer "of the", nous prenons aussi en compte "of" et "the". Avec *custom-2*, pour "The quick brown fox jumps", nous avons "The quick brown", "quick brown fox" et "brown fox jumps". Ceci entraîne alors une augmente rapide du nombre de termes.

Concernant les 10 premiers termes de chaque analyzer, on remarque que *whitespace* et *custom-1* contiennent une majorité de stopwords vu que ces derniers ne sont pas filtrés. *Custom-stop comporte moins de stopwords usuels* car qu'il travaille avec une liste de termes qui sont plus spécifiques comme "system" ou "computer". *English* contient naturellement des termes (parfois composé d'un bout de mot) très courants en anglais. Finalement, pour les analyzers avec shingles, *custom-2*, n'indexe que des shingles d'une taille de trois et custom-1 a un mélange de shingles d'une taille de deux et d'unigrammes, comme voulu par les contraintes que nous avions imposées.

2.5 Searching

D.12

1. Publications containing the term "Information Retrieval"

```
GET cacm_english/_search
{
  "stored_fields": ["id"],
  "query": {
    "query_string": {
      "query": "\"Information Retrieval\"",
      "fields": ["summary"]
    }
  }
}
```

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

```
GET cacm_english/_search
{
  "stored_fields": ["id"],
  "query": {
    "query_string": {
      "query": "(Information) AND (Retrieval)",
      "fields": ["summary"]
    }
  }
}
```

3. Publications containing at least the term "Retrieval" and, possibly "Information" but not "Database".

```
GET cacm_english/_search
{
  "stored_fields": ["id"],
  "query": {
    "query_string": {
      "query": "+Retrieval Information -Database",
      "fields": ["summary"]
    }
  }
}
```

4. Publications containing a term starting with "Info".

```
GET cacm_english/_search
{
  "stored_fields": ["id"],
  "query": {
    "query_string": {
      "query": "Info*",
      "fields": ["summary"]
    }
  }
}
```

5. Publications containing the term "Information" close to "Retrieval" (max distance 5).

```
GET cacm_english/_search
{
  "stored_fields": ["id"],
  "query": {
    "query_string": {
      "query": "\"Information Retrieval\"~5",
      "fields": ["summary"]
    }
  }
}
```

D.13

1. Publications containing the term "Information Retrieval"

```
"hits" : {
    "total" : {
      "value" : 20,
```

```
      "relation" : "eq"
    },
```

2. Publications containing both “Information” and “Retrieval”

```
"hits" : {
    "total" : {
      "value" : 36,
      "relation" : "eq"
    }
```

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

```
"hits" : {
    "total" : {
      "value" : 69,
      "relation" : "eq"
    }
```

4. Publications containing a term starting with “Info”.

```
"hits" : {
    "total" : {
      "value" : 205,
      "relation" : "eq"
    }
```

5. Publications containing the term “Information” close to “Retrieval” (max distance 5).

```
"hits" : {
    "total" : {
      "value" : 30,
      "relation" : "eq"
    },
```

2.6 Tuning the Lucene Score

2.6.1 Custom similarity

D.14

```
PUT cacm_custom-score
{
  "settings": {
    "index": {
      "similarity": {
        "default": {
          "type": "scripted",
          "script": {
            "source": "double tf = 1.0 + Math.log(doc.freq); double idf = Math.log(field.docCount/(term.docFreq+1.0)) + 1.0; double norm = 1.0; return query.boost * tf * idf * nor
          }
        }
      }
    },
    "analysis": {
      "analyzer": {
        "default": {
          "type": "standard"
        }
      }
    }
  },
  "mappings": {
    "properties": {
      "id": {
        "type": "keyword",
        "index": false,
        "store": true
      },
      "author": {
        "type": "keyword"
      },
      "title": {
        "type": "text",
        "fielddata": true
      },
      "date": {
        "type": "date"
      },
      "summary": {
        "type": "text",
        "fielddata": true,
        "index_options": "offsets"
      }
    }
  }
}

POST _reindex
{
  "source": {
    "index": "cacm_dynamic"
  },
  "dest": {
    "index": "cacm_custom-score"
  }
}
```

D.15

#	ids with custom scoring	Score with custom scoring	ids without custom scoring	Score without custom scoring
1	2534	14.1800995	3189	7.9927473
2	1647	12.428032	123	7.0958896
3	1739	11.304348	678	7.0101233
4	2423	11.304348	1465	6.926655
5	123	11.234488	1739	6.7396317
6	678	11.234488	1647	6.4866796
7	1465	11.234488	2534	6.4402065
8	3189	10.190798	1676	6.2567477
9	1215	9.900334	2423	6.1307526
10	2897	9.900334	598	6.0692043

```
GET cacm_standard/_search
{
  "stored_fields": ["id"],
  "query": {
    "query_string": {
      "query": "compiler program"
    }
  }
}
```

2.6.2 Function score

D.16

```
GET cacm_standard/_search
{
  "query": {
    "function_score": {
      "query": {
        "query_string": {
          "query": "compiler program"
        }
      },
      "linear": {
        "date": {
          "origin": "1970-01",
          "scale": "90d",
          "decay": 0.5
        }
      }
    }
  }
}
```