Rapport Lab03 elasticSearch

D1

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

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

D2

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

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

D3

```
GET /cacm_termvector/_termvectors/G2hBV44BbRQEeoKD6n1I
```

```
{
  "_index": "cacm_termvector",
  "_id": "G2hBV44BbRQEeoKD6n1I",
```

```
"_version": 1,
"found": true,
"took": 2,
"term_vectors": {
  "summary": {
    "field_statistics": {
      "sum_doc_freq": 97730,
      "doc_count": 1585,
      "sum_ttf": 150220
    },
    "terms": {
      "a": {
        "term_freq": 1
      },
      "accelerates": {
        "term_freq": 1
      },
      "an": {
        "term_freq": 3
      },
      "and": {
        "term_freq": 1
      },
      "applied": {
        "term_freq": 1
      },
      "convergence": {
        "term_freq": 2
      },
      "converges": {
        "term_freq": 1
      },
      "discussed": {
        "term_freq": 1
      },
      "diverges": {
        "term_freq": 1
      },
      "equation": {
        "term_freq": 1
      },
      "example": {
        "term_freq": 1
      },
      "for": {
        "term_freq": 1
      },
      "given": {
        "term_freq": 1
      },
      "if": {
        "term_freq": 2
      },
```

```
        "illustrative": {
          "term_freq": 1
        },
        "induces": {
          "term_freq": 1
        },
        "is": {
          "term_freq": 2
        },
        "iteration": {
          "term_freq": 2
        },
        "iterative": {
          "term_freq": 1
        },
        "of": {
          "term_freq": 2
        },
        "procedure": {
          "term_freq": 1
        },
        "rate": {
          "term_freq": 1
        },
        "solution": {
          "term_freq": 1
        },
        "technique": {
          "term_freq": 1
        },
        "the": {
          "term_freq": 4
        },
        "to": {
          "term_freq": 1
        },
        "when": {
          "term_freq": 1
        },
        "which": {
          "term_freq": 1
        }
      }
    }
  }
}
```


D4

In Elasticsearch, a term vector is a data structure that provides detailed information about the terms (words) within a specific field of a document. Term vectors contain statistics about each term, such as their frequency, positions, offsets, and more. They are useful for various tasks, including document analysis, relevance scoring, and text mining.

D5

```
GET /_cat/indices/cacm_standard,cacm_termvector?v
```

```
health status index           uuid                   pri rep docs.count docs.deleted store.siz
yellow open   cacm_termvector MbFLDQdmTUWCdf4eFT2ObA   1   1       3202            0      2.1m
yellow open   cacm_standard   U1GkWjLdRgC5z5U5uoEu1w   1   1       3202            0      1.6m
```

The cacm_standard index has a smaller store size than the cacm_termvector index, which is expected since the term vector index stores more information about the terms in the documents. The term vector index has a store size of 2.1mb, while the standard index has a store size of 1.6mb.

D6

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

```
{
  "took": 164,
  "timed_out": false,
  "_shards": {
    "total": 1,
    "successful": 1,
    "skipped": 0,
    "failed": 0
  },
  "hits": {
    "total": {
```

```
      "value": 3202,
      "relation": "eq"
    },
    "max_score": null,
    "hits": []
  },
  "aggregations": {
    "authors": {
      "doc_count_error_upper_bound": 0,
      "sum_other_doc_count": 3083,
      "buckets": [
        {
          "key": "Thacher Jr., H. C.;",
          "doc_count": 36
        }
      ]
    }
  }
}
```


The author with the most documents in the cacm_standard index is "Thacher Jr., H. C.;", with 36 documents.

D7

```
GET /cacm_standard/_search
{
  "size": 0,
  "aggs": {
    "top_titles": {
      "terms": {
        "field": "title",
        "size": 10,
        "order": {
          "_count": "desc"
        }
      }
    }
  }
}
```


```
{
  "took": 232,
  "timed_out": false,
  "_shards": {
    "total": 1,
    "successful": 1,
    "skipped": 0,
```

```
    "failed": 0
  },
  "hits": {
    "total": {
      "value": 3202,
      "relation": "eq"
    },
    "max_score": null,
    "hits": []
  },
  "aggregations": {
    "top_titles": {
      "doc_count_error_upper_bound": 0,
      "sum_other_doc_count": 17307,
      "buckets": [
        {
          "key": "of",
          "doc_count": 1138
        },
        {
          "key": "algorithm",
          "doc_count": 975
        },
        {
          "key": "a",
          "doc_count": 895
        },
        {
          "key": "for",
          "doc_count": 714
        },
        {
          "key": "the",
          "doc_count": 645
        },
        {
          "key": "and",
          "doc_count": 434
        },
        {
          "key": "in",
          "doc_count": 416
        },
        {
          "key": "on",
          "doc_count": 340
        },
        {
          "key": "an",
          "doc_count": 275
        },
        {
          "key": "computer",
```

```
          "doc_count": 275
        }
      ]
    }
  }
}
```

Term	Frequency
of	1138
algorithm	975
a	895
for	714
the	645
and	434
in	416
on	340
an	275
computer	275

D8

Whitespace Analyzer

```
PUT /cacm_whitespace
{
  "settings": {
    "analysis": {
      "analyzer": {
        "default": {
          "type": "whitespace"
        }
      }
    }
  },
  "mappings": {
    "properties": {
      "id": {
        "type": "keyword",
        "store": true,
        "index": false
```

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

```
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",
        "store": true,
        "index": false
```

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


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

Custom (Standard + Shingles 1 & 2) Analyzer

```
PUT /cacm_custom_shingles12
{
  "settings": {
    "analysis": {
      "analyzer": {
        "custom_shingles12": {
          "type": "custom",
          "tokenizer": "standard",
          "filter": ["lowercase", "shingle_filter_12"]
        }
      },
      "filter": {
        "shingle_filter_12": {
          "type": "shingle",
          "max_shingle_size": 2,
          "output_unigrams": true
        }
```

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

```
POST /_reindex
{
  "source": {
    "index": "cacm_dynamic"
  },
  "dest": {
    "index": "cacm_custom_shingles12"
  }
}
```

Custom (Standard + Shingles 1 & 3) Analyzer

```
PUT /cacm_custom_shingles13
{
  "settings": {
    "analysis": {
      "analyzer": {
        "custom_shingles3": {
          "type": "custom",
          "tokenizer": "standard",
```

```
          "filter": ["lowercase", "shingle_filter_3"]
        }
      },
      "filter": {
        "shingle_filter_3": {
          "type": "shingle",
          "max_shingle_size": 3,
          "min_shingle_size": 3,
          "output_unigrams": false
        }
      }
    }
  },
  "mappings": {
    "properties": {
      "id": {
        "type": "keyword",
        "store": true,
        "index": false
      },
      "author": {
        "type": "keyword"
      },
      "title": {
        "type": "text",
        "fielddata": true,
        "analyzer": "custom_shingles3"
      },
      "date": {
        "type": "date"
      },
      "summary": {
        "type": "text",
        "index_options": "offsets",
        "fielddata": true,
        "analyzer": "custom_shingles3"
      }
    }
  }
}


POST /_reindex
{
  "source": {
    "index": "cacm_dynamic"
  },
  "dest": {
    "index": "cacm_custom_shingles13"
  }
}
```

Stopwords Analyzer

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

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

```
    }
  }
```

D9

Whitespace Analyzer

The whitespace analyzer tokenizes the text into terms whenever it encounters a whitespace character. It does not perform any further processing on the terms, such as lowercasing or stemming. This means that the terms are stored as-is, without any modifications. This can be useful for preserving the original structure of the text, but it may not be suitable for all use cases, such as searching for terms with different casing or stemming variations.

English Analyzer

The English analyzer is specifically designed for processing English text. It tokenizes the text into terms, lowercases the terms, removes common English stopwords, and applies stemming to reduce the terms to their root form. This can help improve search accuracy by normalizing the terms and reducing the number of variations that need to be matched. It is well-suited for English language text analysis, but it may not be optimal for other languages or specialized domains.

Custom (Standard + Shingles 1 & 2) Analyzer

The custom analyzer combines the standard tokenizer with a shingle filter that generates shingles (word pairs) of size 1 and 2. This means that each term in the text is broken down into unigrams (single words) and bigrams (word pairs). This can help capture more context and relationships between terms in the text, which can be useful for certain types of analysis, such as phrase matching or proximity search. However, it may increase the index size and query complexity.

Custom (Standard + Shingles 3) Analyzer

The custom analyzer combines the standard tokenizer with a shingle filter that generates shingles of size 3. This means that each term in the text is broken down into trigrams (word triplets). This can capture even more context and relationships between terms compared to shingles of size 1 and 2. It can be useful for analyzing longer phrases or capturing more complex patterns in the text. However, it may further increase the index size and query complexity compared to smaller shingles.

Stopwords Analyzer

The custom analyzer uses a list of English stopwords (found in the `common_words.txt` file) to filter out common words that are not relevant for search or analysis. This can help reduce the index size, improve search performance, and focus on more meaningful terms in the text. However, it may also filter out important terms that happen to be stopwords or cause unexpected behavior if the

stopwords list is not properly curated. It is important to carefully select and maintain the list of stopwords based on the specific use case and domain.

D10

Analyzer	Indexed Documents	Indexed Terms	Top 10 Frequent Terms	Index Size	Indexing Time
Whitespace Analyzer	3202	15653	of, the, is, and, a, to, in, for, The, are	1.7mb	313ms
English Analyzer	3202	5221	which, us, comput, program, system, present, describ, paper, can, gener	1.5mb	346ms
Custom (Shingles 1 & 2)	3202	78925	the, of, a, is, and, to, in, for, are, of the	3.3mb	512ms
Custom (Shingles 3)	3202	123158	in this paper, the use of, the number of, it is shown, a set of, in terms of, the problem of, is shown that, a number of, as well as	3.6mb	400ms
Stopwords Analyzer	3202	7936	computer, system, paper, presented, time, program, data, method, algorithm, discussed	1.4mb	252ms

D11

1. The choice of analyzer can have a significant impact on the indexing process and the resulting index size. Analyzers that generate more terms, such as shingle analyzers, can lead to larger indexes with more terms, which may affect search performance and resource usage. It is important to carefully select an analyzer that balances the need for detailed analysis with the practical considerations of index size and query complexity.
2. Analyzers that perform additional processing steps, such as stopword removal, stemming, or shingling, can help improve search accuracy and relevance by normalizing the terms and capturing more context. However, these additional steps may also introduce complexity and trade-offs in terms of index size, query performance, and maintenance. It is important to evaluate the trade-offs and choose an analyzer that best fits the specific requirements of the use case.

3. The indexing time can vary depending on the complexity of the analyzer and the amount of text being processed. Analyzers that perform more processing steps, such as shingling or stemming, may require more time to index the documents compared to simpler analyzers. It is important to consider the indexing time as part of the overall performance evaluation and optimization process when selecting an analyzer for a specific use case.

D12

1. Publications containing the term “Information Retrieval”.

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

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

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

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

```
GET /cacm_english/_search
{
  "query": {
    "query_string": {
      "query": "summary:(Retrieval) AND NOT summary:(Database)"
    }
  },
  "_source": ["id"]
}
```

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

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

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

```
GET /cacm_english/_search
{
  "query": {
    "match_phrase": {
      "summary": {
        "query": "Information Retrieval",
        "slop": 5
      }
    }
  },
  "_source": ["id"]
}
```

D13

1. Publications containing the term “Information Retrieval”.

```
    "total": {
      "value": 20,
      "relation": "eq"
    }
```

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

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

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


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


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


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


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


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


D14

```
GET /cacm_english/_search
{
  "query": {
    "function_score": {
      "linear": {
        "date": {
          "origin": "1970-01",
          "scale": "90d",
          "decay": 0.5
        }
      },
      "query": {
        "match": {
          "summary": "Information Retrieval"
        }
      }
    }
  }
}
```