CE706 - Information Retrieval SU 2022

Assignment 1

Student Id : 2110376

Instructions for running your system

The project demonstrates the capabilities of elastic search engine as a Query tool for reliable search and analysis. The setup involves the following steps:

Assumptions:

1) The setup instructions assume user operating system is Linux/Ubuntu.
2) The User system is loaded with a Python3 Environment.
3) GitHub access to repository:https://github.com/panks11/CE706.git

Setup:

1) **Downloading Elastic search and Kibana using curl commands:**

   curl -L - O https://artifacts.elastic.co/downloads/elasticsearch/elasticsearch-6.5.1.tar.gz
   curl -L -O https://artifacts.elastic.co/downloads/kibana/kibana-6.5.1-linux-x86_64.tar.gz
2) **Install Jave Runtime Environment:**
   `sudo apt install default-jre`
3) **Run elastic search**
   elasticsearch-6.5.1/bin/elasticsearch -d and
   Kibana *elasticsearch-6.5.1/bin/elasticsearch -d.*
4) To Check, Kibana GUI should open successfully at http://localhost:5601/
5) For Python Elastic Search API installation execute *pip install elasticsearch*
6) Git clone repository : https://github.com/panks11/CE706.git

Indexing

The Signal Media One Million New Article dataset was downloaded after filling a Google Request Form to download the data released by Signal Media on their NewsIR'16 webpage. [Link]. A file 'sample-1M.jsonl' is extracted containing the list of articles. Each article object contains following fields:

id: a unique identification number for the article

title: The article Summary

content: Content of the article

source: Publisher details

published: Date of Publication

media-type: The Type of Article : News or Blog

id	content	title	media-type	source	published
864a0952-aae8-4ee6-b3bb-9b11d010cf43	Queen Sandra RULES! however anyone may comment...	Queen Sandra..... owls/ any color/ spoked eye...	Blog	DeniseAnnette	2015-07-03 19:51:40
fd824d39-9ffc-4b03-acb2-6944bc20cd5f	Happy 4th of July to all our friends and famil...	Happy 4th of July 2015!	Blog	Charlottesville Solutions	2015-07-04 11:41:21
0867cb05-75fd-44b1-ada9-b775c5467d4d	(0 comments - 497 views) \n#1 Video insane cam...	Weekly Achievements for 28Jun15 thru 04Jul15	Blog	Latest Blog Entries at VideoSift.com	2015-07-05 07:01:03
e520a3c3-e5dd-4c1e-877e-da9711f8c938	Nasi kandar in Penang Every now and then you h...	Nasi Kandar Penang: Insanely Good Curry at Taj...	Blog	Migrationology - Food Travel Blog	2015-07-05 13:00:32
c7a71b47-eb09-43de-a222-81cf40a2190d	Nasi kandar in Penang Every now and then you h...	Nasi Kandar Penang: Insanely Good Curry at Taj...	Blog	Migrationology - Food Travel Blog	2015-07-05 13:00:32

Figure 1: Sample Content of 'sample-1M.jsonl'

The Articles are majorly of 'News' or 'Blog' type and published latest in the year of '2015'. The publishers of these articles include majorly Reuters and other local news sources. The file contains 1 million multi lingual articles mainly in English. The uncompressed dataset file exceeds 2.5 GB of file and cannot be uploaded to Elastic Search at once. Therefore, I have read the json file using Python and **used Python Elastic Search API** to post the documents using es.index() method.

Challenges:

1. Huge size of dataset hampered the direct methods to upload the document using REST API via curl or Kibana
2. The Signal dataset documents in the 'jsonl' file did not contain indexes. For Elastic search upload format expect the documents to be tagged with an index.
3. The escape sequences and unicode characters disturb the format of an expected JSON file and thus elastic search Bulk API method threw Syntax errors.

To overcome the challenges, Python library to read JSON was used and Elastic Search API es.index() method was used to post documents iteratively. To select the subset, I have sorted the documents according to their published date and posted 100000

```
GET /_cat/indices?v

1 health status index     uuid                   pri rep docs.count docs.deleted store.size pri.store.size
2 yellow open   articles  BKs902NfTPGYDr-H6xJ08w   5   1     112168            0    328.6mb        328.6mb
3 green  open   .kibana_1 8tfsLIdYSLuxbq6vgeQsyA   1   0         16            1     75.2kb         75.2kb
4
```

Figure 1.1 Document Count for articles

Improvements:

An improved way to push data in Elastic search would be to involve FileBeat which parses and import data in near real time or Logstash which can push data into Elastic Search from many machines.

Indexing and Mapping:

The elastic search implicitly does the field mapping after understanding the nature of fields, however, I have created the mappings explicitly according to the type of fields as shown in figure. As the default mappings for all the fields were mapped to 'text', I have explicitly mapped the '**published**' field to 'Date'. The fields **id**, **content** and **source** are mapped to text, and **title** and **content** to text.

```
PUT articles
{
  "settings": {
    "analysis": {
      "analyzer": {
        "my_english_analyzer": {

          "tokenizer": "uax_url_email",
          "filter": [
            "lowercase",
            "asciifolding",
            "english_stop",
            "porter_stem"

          ]
        }
      },
      "filter": {
        "english_stop": {
          "type": "stop",
          "stopwords": "_english_"
        }
      }
    }
  },
  "mappings": {
    "news":{
    "properties": {
      "id":     { "type": "keyword" },
      "title" : {"type":"text"},
      "content" : {"type":"text"},
      "source" : {"type":"keyword"},
      "published":    { "type": "date"},
      "media-type":  { "type": "keyword"  }
}}
}

}
```

Figure 2 : Mapping

Tokenization and Normalisation

Tokenization is breaking a stream of characters into individual tokens usually words. ElasticSearch offer a number of built in tokenizers like standard, Letter, whitespace etc. The builtin tokenizers can be used to create custom analyzers. Additinaly the tokenizer records the position of each term, start and end character offsets and token type to identify the term produced as Alphanum, Num etc. I have a **custom analyzer** to implement 'Standard' Tokenizer which was able to divide terms on word boundaries and remove punctuations. For reference, Figure 3. show a sample of Tokenization, where the terms are generated successfully with word boundaries. The special characters were ignored successfully and punctuations are removed.

Figure 3. 'Standard' Tokenization

However, Standard Tokenizer is not smart enough to handle email, Url and Timestamps as shown in Figure 4 The terms of the email are separated incorrectly.

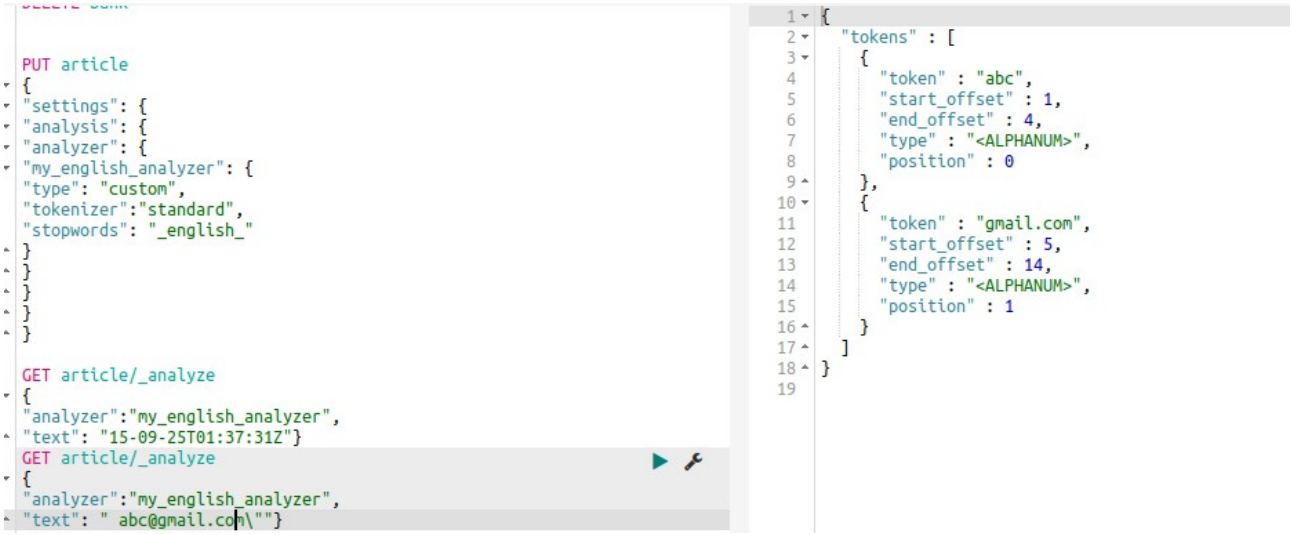

Figure 4: Tokenization Challenges

I have used **uax_url_email** tokenizer which is similar to standard tokenizer except that it can recognise URLs and email as single tokens.

```
PUT article
{
"settings": {
"analysis": {
"analyzer": {
"my_english_analyzer": {
"type": "custom",
"tokenizer":"uax_url_email",
"stopwords": "_english_"
}
}
}
}
}

GET article/_analyze
{
"analyzer":"my_english_analyzer",
"text": " abc@gmail.com\""}
```

```
{
  "tokens" : [
    {
      "token" : "abc@gmail.com",
      "start_offset" : 1,
      "end_offset" : 14,
      "type" : "<EMAIL>",
      "position" : 0
    }
  ]
}
```

Figure 5: uax_url_email Tokenizer

I have applied case folding using **token filters** to convert the words into lowercase. The case folding helps in reducing the vocabulary size by reducing the difference in same words and different case.

```
PUT article
{
"settings": {
"analysis": {
"analyzer": {
"my_english_analyzer": {
"type": "custom",
"tokenizer":"uax_url_email",
"stopwords": "_english_",
"filter": [ "porter_stem",
            "lowercase",
            "asciifolding"
          ]
}
}
}
}
}

GET article/_analyze
{
"analyzer":"my_english_analyzer",
"text": " My Email is abc@gmail.com\""}
```

```
{
  "tokens" : [
    {
      "token" : "my",
      "start_offset" : 1,
      "end_offset" : 3,
      "type" : "<ALPHANUM>",
      "position" : 0
    },
    {
      "token" : "email",
      "start_offset" : 4,
      "end_offset" : 9,
      "type" : "<ALPHANUM>",
      "position" : 1
    },
    {
      "token" : "is",
      "start_offset" : 10,
      "end_offset" : 12,
      "type" : "<ALPHANUM>",
      "position" : 2
    },
    {
      "token" : "abc@gmail.com",
      "start_offset" : 13,
      "end_offset" : 26,
      "type" : "<EMAIL>",
      "position" : 3
    }
  ]
}
```

Figure 6 : Case Folding

For Normalization I have used **ASCII Folding** which converts the alphabetic, numeric, and symbolic characters into their ASCII equivalent.

Selecting Keywords

Elastic search provides removal of Stop words using a **'Stop token filter'.** I have added a stop word filter to my customized analyzer to remove stop words of English. The stop words do not add much information to the sentence and it is a good practice to remove the stop words to reduce the vocabulary size and improve search performance. Figure 7 shows the English stop word list which will be removed by the analyzer as shown in Figure 8.

```
"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");
```

Figure 7: English Stop Word List

```
PUT article
{
  "settings": {
    "analysis": {
      "analyzer": {
        "my_english_analyzer": {

          "tokenizer": "uax_url_email",
          "filter": [
            "lowercase",
            "english_stop"
          ]
        }
      },
      "filter": {
        "english_stop": {
          "type": "stop",
          "stopwords": "_english_"
        }
      }
    }
  }
}

GET article/_analyze
{
"analyzer":"my_english_analyzer",
"text": " The fox is in the jungle"

}
```

```
{
  "tokens" : [
    {
      "token" : "fox",
      "start_offset" : 5,
      "end_offset" : 8,
      "type" : "<ALPHANUM>",
      "position" : 1
    },
    {
      "token" : "jungle",
      "start_offset" : 19,
      "end_offset" : 25,
      "type" : "<ALPHANUM>",
      "position" : 5
    }
  ]
}
```

Figure 8: Stop Word Removal

The boolean model in Elastic Search is similar to a binary search and excludes the documents which do not qualify and does not care about relevancy. For improving relevancy of our search results we can combine TF/IDF and Boolean model to rank the documents. The Vector space model is used to compare multi term queries in a document by calculating a similarity score per field on vectors, Term Frequency i.e. How often a term appears in a document and Inverse Document Frequency i.e. How often a term appears in an index. I have used the 'Scripted' Similarity to apply the TF-IDF for selection and weighting step.

```
body = ("""{
  "settings": {
    "similarity": {
      "scripted_tfidf": {
        "type": "scripted",
        "script": {
          "source": "double tf = Math.sqrt(doc.freq); double idf = Math.log((field.docCount+1.0)/(term.docFreq+1.0)) + 1.0; double norm = 1/Math.sqrt(doc.length);
          return query.boost * tf * idf * norm;"
        }
      }
    },
    "analysis": {
      "analyzer": {
        "my_english_analyzer": {

          "tokenizer": "uax_url_email",
          "filter": [
            "lowercase",
            "asciifolding",
            "english_stop",
            "porter_stem"

          ]
        }
      },
      "filter": {
        "english_stop": {
          "type": "stop",
          "stopwords": "_english_"
        }
      }
    }
  },
  "mappings": {
    "news":{
    "properties": {
      "id":    { "type": "keyword" },
      "title" : {"type":"text", "similarity":"scripted_tfidf"},
      "content" : {"type":"text", "similarity":"scripted_tfidf"},
      "source" : {"type":"keyword"},
      "published":   { "type": "date"},
      "media-type":  { "type": "keyword"  }
  }}
}

}""")
```

Figure 8.1 Similarity API Method

Stemming or Morphological Analysis

Stemming can be explained as reducing a word to its root form. This allows the search of variants of a root word. The scope of the search is not limited to a specific word and can be matched to its variants as well. For instance, root word for running and run can be stemmed to same root word: run. I have used algorithmic stemmers as they require less memory and setup to achieve good results. They are also fast in execution in comparison to dictionary stemmers. I have used **porter_stemmer** which is a recommended stemmer for English language. Since it requires lowercased words to work properly, I have first added the lowercase and then the stemmer in my custom analyzer.

PUT article

{ "settings": { "analysis": {

"analyzer": {

"my_english_analyzer": {

"tokenizer": "uax_url_email",

"filter": ["lowercase", "english_stop", "porter_stem"]}},

"filter": {"english_stop": { "type": "stop", "stopwords": "_english_"}}}

}

}

I have added a screenshot of sample data from our dataset and highlighted a stemmed example.

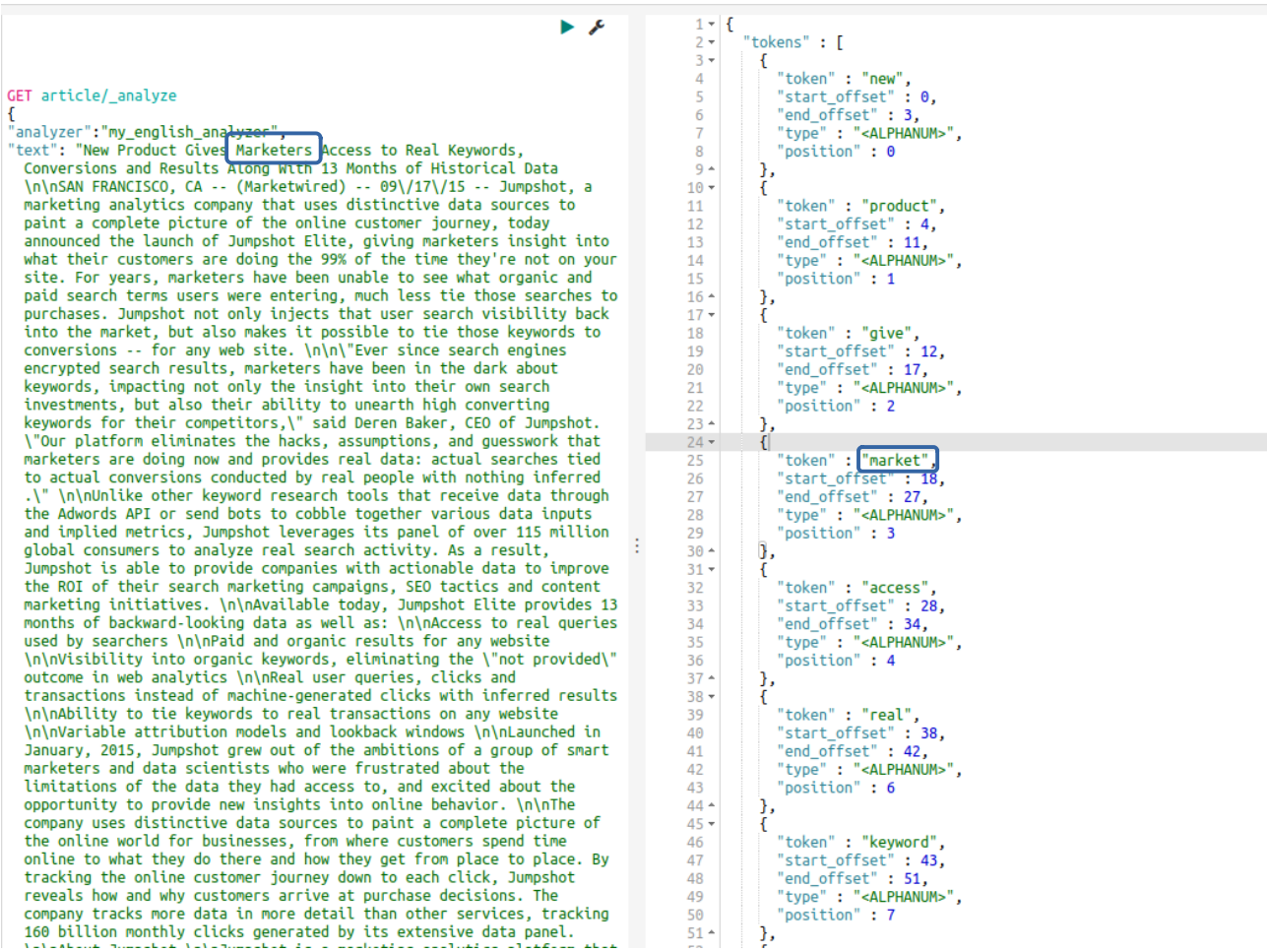

Figure 8: Porter Stemming

Searching

Query 1:

Find all articles of a given media-type for a specific publisher and sort them according to their Publish Date.

The query is composed of filtering and sorting. It matches Media Type and filters the source field using the User Input. The sorting is done on published field.

Here, since articles index fields media-type and source are mapped to ‘keywords’, Matching Part of Phrases is not working. The fields can be mapped to ‘text’ for more flexible search.

```
GET /articles/_search
{
"query": {
"bool": {
"must": {
"match": {
"media-type": "News"
}
},
"filter": {
"term": {
"source": "4 Traders"
}
}
}
},
"sort": { "published": { "order": "desc" } }
}
```

```
{
  "took" : 1,
  "timed_out" : false,
  "_shards" : {
    "total" : 5,
    "successful" : 5,
    "skipped" : 0,
    "failed" : 0
  },
  "hits" : {
    "total" : 347,
    "max_score" : null,
    "hits" : [
      {
        "_index" : "articles",
        "_type" : "news",
        "_id" : "dd7f96a6-18ab-47fe-ab88-9620e989f56d",
        "_score" : null,
        "_source" : {...},
        "sort" : [
          1443651768000
        ]
      },
```

Query 2:

Find the total number of Blogs and News type articles in the dataset

The query use Filter aggregations, where a multi bucket aggregation is created using a Filter.

The two buckets consist of ‘News’ and ‘Blog’ media type.

```
GET articles/_search
{
  "size":0,
  "aggs": {
    "Articles": {
      "filters": {
        "filters": {
          "Blog": {"term":{"media-type": "Blog"}},
          "News":{"term":{"media-type":"News"}}
        }
      }
    }
  }
}
```

```
{
  "took" : 2,
  "timed_out" : false,
  "_shards" : {
    "total" : 5,
    "successful" : 5,
    "skipped" : 0,
    "failed" : 0
  },
  "hits" : {
    "total" : 99999,
    "max_score" : 0.0,
    "hits" : [ ]
  },
  "aggregations" : {
    "Articles" : {
      "buckets" : {
        "Blog" : {
          "doc_count" : 26661
        },
        "News" : {
          "doc_count" : 73338
        }
      }
    }
  }
}
```

Query 3:

Find relevant documents matching a User defined query boosting the search using TF-IDF score.

The query searches all the articles related to Scotland:

```
GET articles/_search?explain=true
{
  "query": {
    "query_string": {
      "query": "Scotland^1.7",
      "default_field": "content"
    }
  }
}
```

```
 Scotland can still reach the Euro 2... 3 minutes ago
""",
        "title" : "Scotland can still qualify - Strachan",
        "media-type" : "News",
        "source" : "NewsR.in",
        "published" : "2015-09-07T22:58:07Z"
      },
      "_explanation" : {
        "value" : 2.7630956,
        "description" : "weight(content:scotland in 4660) [PerFieldSimilarity], result of:",
        "details" : [
          {
            "value" : 2.7630956,
            "description" : "score from ScriptedSimilarity(weightScript=[null], script=[Script{type=inline, lang='painless'
              , idOrCode='double tf = Math.sqrt(doc.freq); double idf = Math.log((field.docCount+1.0)/(term.docFreq+1.0)) +
              1.0; double norm = 1/Math.sqrt(doc.length); return query.boost * tf * idf * norm;', options={}, params={}}])
              computed from:",
            "details" : [
              {
                "value" : 1.0,
                "description" : "weight",
                "details" : [ ]
              },
              {
                "value" : 1.7,
                "description" : "query.boost",
                "details" : [ ]
              },
              {
                "value" : 21330.0,
                "description" : "field.docCount",
                "details" : [ ]
              },
              {
                "value" : 4221970.0,
                "description" : "field.sumDocFreq",
                "details" : [ ]
              },
              {
                "value" : 8477385.0,
                "description" : "field.sumTotalTermFreq",
                "details" : [ ]
              },
              {
                "value" : 207.0,
                "description" : "term.docFreq",
                "details" : [ ]
              },
              {
                "value" : 515.0,
                "description" : "term.totalTermFreq",
                "details" : [ ]
              },
```

Challenge:

It was difficult to identify non English documents in the huge dataset. The Text preprocessing steps like tokenization, stop word removal are done majorly considering ‘English’ as main language.

Improvements:

Using NLTK library language detection model the language of the articles can be detected and text pre processing steps can be configured accordingly.

Elastic Search API:

The above mentioned operations have been developed in Python using the Python Client API for Elastic Search.

```
pk@pkvm:~/Documents/lr/search_exercise/assignment/CE70$ python3 get.py
{'name': 'Y7eRIS7', 'cluster_name': 'elasticsearch', 'cluster_uuid': '5qp3cRqIR7S5tYvmAdKA0Q', 'version': {'number': '6.5.1', 'build_flavor': 'default', 'build_type': 'tar', 'build_hash': '8c58350', 'bui
ld_date': '2018-11-16T02:22:42.182257Z', 'build_snapshot': False, 'lucene_version': '7.5.0', 'minimum_wire_compatibility_version': '5.6.0', 'minimum_index_compatibility_version': '5.0.0'}, 'tagline': 'Yo
u Know, for Search'}
*******************Query 1 Results ****************************
Score None
{'id': 'dd7f96a6-18ab-47fe-ab88-9620e989f56d', 'content': 'All dollar amounts are in Canadian dollars. \n \nSAINT-GEORGES, QC , Sept. 30, 2015 /CNW Telbec/ - Manac Inc. (TSX: MA) ("Manac" or the "Company
"), a North American leader in the design and manufacture of specialty trailers, announced today that its holders of multiple voting shares and subordinate voting shares have approved the resolution auth
orizing the previously announced statutory arrangement under the Business Corporations Act ( Québec ) (the\xa0"Arrangement") pursuant to which a consortium (the\xa0"Consortium") composed of Placements CM
I Inc. ("CMI"), Caisse de dépôt et placement du Québec ("CDPQ"), Fonds de solidarité FTQ ("FSTQ"), Investissement Québec and Fonds Manufacturier Québécois II s.e.c. will indirectly acquire all of the iss
ued and outstanding multiple voting shares and subordinate voting shares for a cash consideration of $10.20 per share. \n \nThe Arrangement resolution required the approval of at least 66 2/3 % of the vo
tes cast by holders of multiple voting shares and subordinate voting shares present in person or represented by proxy at the special meeting of shareholders held earlier today (the\xa0"Special Meeting"),
 voting together as a single class, with each holder being entitled to one vote per share. Given that the proposed Arrangement constitutes a "business combination" under Regulation 61-101 respecting Prot
ection of Minority Security Holders in Special Transactions , it was also subject to the approval of (i)\xa0a majority of the votes cast by the holders of multiple voting shares (excluding CMI and LITUD
Inc. ("LITUD"), a holding corporation controlled by Mr. Charles Dutil ) present in person or represented by proxy at the Special Meeting, and (ii)\xa0a majority of the votes cast by the holders of subord
inate voting shares (excluding CDPQ and FSTQ) present in person or represented by proxy at the Special Meeting, each voting as a separate class. \n \nThe Arrangement resolution was approved by 99.995% of
 the votes cast by holders of multiple voting shares and subordinate voting shares present in person or represented by proxy at the Special Meeting, voting together as a single class, with each holder be
ing entitled to one vote per share, and by\xa0(i) 100% of the votes cast by the holders of multiple voting shares (excluding CMI and LITUD) present in person or represented by proxy at the Special Meetin
g, and (ii)\xa099.969% of the votes cast by the holders of subordinate voting shares (excluding CDPQ and FSTQ) present in person or represented by proxy at the Special Meeting, each voting as a separate
class. \n \nThe implementation of the Arrangement is subject to approval by the Québec Superior Court at a final hearing which is scheduled to be held on October 5, 2015 at the Saint-Joseph-de-Beauce Cou
rthouse at 10:30 a.m. (Montreal\xa0Time). It is currently anticipated that the Arrangement will be completed in October 2015 , subject to, without limitation, approval by the Québec Superior Court as set
 forth above and the satisfaction or waiver of the other conditions precedent to the Arrangement. Further details regarding the Arrangement are set out in the management information circular dated August
 28, 2015 which is available under the profile of Manac at www.sedar.com. \n \nAbout Manac Inc. \n \nManac is the largest manufacturer of trailers in Canada and a leader in the manufacturing of specialty
 trailers in North America . Manac offers a wide range of vans, flatbeds and specialty trailers such as dumps, low beds, grain hoppers, chassis, chip and logging trailers, all of which are sold in Canada
 and the United States under the recognized brands Manac®, CPS®, Peerless®, Darkwinge, UltraPlate®, Ultravan TM and Liddell Canada®. Manac services the heavy-duty trailer industry for the highway transpo
rtation, construction, energy, mining, forestry and agricultural sectors and manufactures its trailers in facilities located in Saint-Georges, QC , Penticton, BC as well as Oran and Kennett, MO. \n \nFor
ward looking statements \n \nCertain statements set forth in this press release may constitute forward-looking statements within the meaning of securities legislation. Positive or negative verbs such as
"believe", "could", "should", "intend", "expect", "estimate", "assume" and other related expressions are used to identify such statements. These forward-looking statements include, but are not limited to
, statements relating to Manac\'s expectations with respect to the timing and outcome of the proposed Arrangement with the Consortium, court approval and the ability of the parties to the arrangement agr
eement to complete the Arrangement. There can be no assurance that the proposed Arrangement will be completed, or that it will be completed on the terms and conditions contemplated in this press release.
 The proposed Arrangement could be modified or terminated. Accordingly, readers should not place undue reliance on the forward-looking statements and information contained in this press release. \n \nSin
ce forward-looking statements and information address future events and conditions, by their very nature they involve inherent risks and uncertainties. Actual results could differ materially from those c
urrently anticipated due to a number of factors and risks. Readers are cautioned that the foregoing list of factors is not exhaustive. Additional information on other factors that could affect the operat
ions or financial results of Manac, which could in turn also impact the completion of the Arrangement, are described in details in the reports filed from time to time by Manac with securities authorities
 in Canada . \n \nThe forward-looking statements contained in this news release are made as of the date of this release and, accordingly, are subject to change after such date. Unless otherwise required
by applicable securities laws, Manac disclaims any intention or obligation to update or revise any forward-looking statements, whether as a result of new information, future events or otherwise. The forw
ard-looking information in this news release is based on information available as of the date of the release. \n \nSOURCE Manac Inc. \n\no Canada Newswire, source Canada Newswire English', 'title': 'MANA
C : shareholders approve privatization by a group of Québec investors led by the Dutil family', 'media-type': 'News', 'source': '4 Traders', 'published': '2015-09-30T22:22:48Z'}
*******************Query 2 Results ****************************
{'took': 4, 'timed_out': False, '_shards': {'total': 5, 'successful': 5, 'skipped': 0, 'failed': 0}, 'hits': {'total': 117542, 'max_score': 0.0, 'hits': []}, 'aggregations': {'Articles': {'buckets': {'Bl
og': {'doc_count': 31471}, 'News': {'doc_count': 86071}}}}}
*******************Query 3 Results ****************************
Score 2.7637303
{'id': '4b3b89c0-cee8-4633-946f-062abf0450e0', 'content': 'Its A Funny Old Game \xa0 - Gordon Strachan insists Scotland can still reach the Euro 20... http://t.co/6A6D24aCSx #iafog \xa0 36 seconds ago \n
\nFootball5o2S \xa0Scotland can still qualify - Strachan\nhttp://t.co/mii9besZtu\xa0 58 seconds ago \n\nNur Muhamad Fadila \xa0Scotland can still qualify - Strachan: Gordon Strachan insists Scotland can
still reach the Euro 2016 finals d... http://t.co/VwZDmus8Vh\xa0 2 minutes ago \n\nNews Scotland \xa0Scotland can still qualify - Strachan: Gordon Strachan insists Scotland can still reach the Euro 2016
finals d... http://t.co/FAZur7uau9\xa0 2 minutes ago \n\nSport Right Now \xa0Scotland can still qualify - Strachan http://t.co/axsy3ShhIv #bbc \xa0 2 minutes ago \n\nInternet Magazine \xa0BBCi Sports ^|
http://t.co/eEKPQl3RY3 | Scotland can still qualify - Strachan Gordon Strachan insists Scotland can still reach the Euro 2…\xa0 3 minutes ago', 'title': 'Scotland can still qualify - Strachan', 'media-ty
pe': 'News', 'source': 'NewsR.in', 'published': '2015-09-07T22:58:07Z'}
```