Report for COMP6490 Lab1

u5976992, Longfei Zhao, 08/08/2017

Q2

I think `P_5` is one of measures that is suitable for this scenarios. Firstly, normal users just look the first page when they use a search engine online. Secondly, I did a little analysis about the `gov.qrels` , as shown in below. As we can see, for most qrels, there are just not more that 5 relevant document, some of qrels even just own 1 relevant document. Therefore I think P_5 is more reasonable.

```
qrels: 42 relevant :1 not relevant: 132
qrels: 50 relevant :1 not relevant: 186
qrels: 24 relevant :1 not relevant: 208
qrels: 26 relevant :3 not relevant: 181
qrels: 22 relevant :1 not relevant: 230
qrels: 47 relevant :5 not relevant: 207
qrels: 44 relevant :4 not relevant: 173
qrels: 48 relevant :1 not relevant: 211
qrels: 28 relevant :2 not relevant: 146
qrels: 43 relevant :6 not relevant: 167
qrels: 41 relevant :1 not relevant: 169
qrels: 1 relevant :5 not relevant: 136
qrels: 2 relevant :2 not relevant: 126
qrels: 4 relevant :4 not relevant: 149
qrels: 7 relevant :3 not relevant: 179
qrels: 6 relevant :1 not relevant: 187
qrels: 9 relevant :1 not relevant: 212
qrels: 10 relevant :1 not relevant: 180
qrels: 39 relevant :1 not relevant: 238
qrels: 38 relevant :4 not relevant: 188
qrels: 14 relevant :1 not relevant: 174
qrels: 16 relevant :7 not relevant: 157
qrels: 19 relevant :2 not relevant: 166
qrels: 18 relevant :1 not relevant: 122
qrels: 31 relevant :1 not relevant: 157
qrels: 37 relevant :2 not relevant: 167
qrels: 36 relevant :6 not relevant: 142
qrels: 35 relevant :5 not relevant: 203
qrels: 34 relevant :1 not relevant: 109
qrels: 33 relevant :4 not relevant: 191
qrels: 32 relevant :30 not relevant: 182
```

Q3

At the first version, the result of `P_5` is `0.08`. I used `trec_eval -q` to see each query performance. I noticed that qrels_33 has 30 relevant document which is much larger than others but the P_5 of qrels_33 is 0. I think it did very badly. I want to improve this problem.

Q4

I think stemming terms, case-folding and lemmatization will largely improve Elasticsearch's performance. Because the `gov.topics` have many different format of terms, such as capitalization, plural and so on, meanwhile, the initial analyzer `es_search_sample.py` is very simple and not include those things.

Q5

Firstly, I want to change search function with different parameters, such as operator `or` or `and`, using `must` or not. I find that the `search()` performs better than some functions provided in `es_search_sample`. Therefore, I use that in my final code, as below.

```
matches = search(query, es_conn, config.INDEX_NAME)
```

Secondly, with reading the documentation of analyzer, I tried to add or change some filters and tokenizers to `es_search_sample.json`, such as `lowercase`, `keyword`. I found that language analyzers are very powerful and easy to use. Therefor I write a analyzer based on English analyzer in `es_settings_english.json`, as below.

```
{
  "settings": {
    "analysis": {
      "filter": {
        "custom_stems" : {
          "type" : "stemmer_override",
          "rules" : [
            "Coastal => wildlife",
            "Conservancy => conservation",
            "conservancy => conservation",
            "stemmer => stemmer"
          ]
        },
        "english_stop": {
          "type":       "stop",
          "stopwords":  "_english_"
        },
        "english_stemmer": {
          "type":       "stemmer",
          "language":   "english"
        },
```

```
          "english_possessive_stemmer": {
            "type":       "stemmer",
            "language":   "possessive_english"
          }
        },
        "analyzer": {
          "english": {
            "tokenizer":  "standard",
            "filter": [
              "custom_stems",
              "english_possessive_stemmer",
              "lowercase",
              "english_stop",
              "english_stemmer"
            ],
            "char_filter": [
              "html_strip"
            ]
          }
        }
      }
    }
}
```

Q6

After all changes, the `P_5` increase from the begin `0.08` to `0.1161`. Generally, I think the result performs well. Unfortunately, the `P_5` of qrels_33 is still `0`. In fact, I analysed a specific document, named `G00-00-2853860` which is relevant to qrels_33. I writed a little bit `custom_stems` to want to retrieve this document. However it fails.