TP7 - Analyse de tweets 2

Analyse de sentiment appliquée aux tweets - PART 2
MongoDB, ElasticSearch et Kibana

Nicolas Dugué - Université d'Orléans - M2 MIAGE option SIR - 2014

A travers ce second TP, nous allons utilisé une image docker prête à être utilisée contenant MongoDB et ElasticSearch. Kibana sera utilisée sur la machine non virtualisée.

A rendre

Un export du Dashboard ("Save > export schema") ainsi que vos requêtes MongoDB, ElasticSearch doivent être envoyées à nicolas.dugue@univ-orleans.fr

Installation MongoDB + ElasticSearch

Récupération de l'image Docker et lancement d'un container nommé TpTwitter qui redirige les ports MongoDB et ElasticSearch sur votre machine :

```
docker pull nicolasdugue/infraprod2
docker run -i -t -p 27017:27017 -p 28017:28017 -p 9200:9200 -p 9300:9300 --name
tpTwitter nicolasdugue/infraprod2:latest /bin/bash
service ssh start
```

A l'intérieur du container docker, lancer la commande suivante pour démarrer le serveur SSH :

```
service ssh start
```

Puis revenir sur le terminal de votre machine et lancer la commande qui suit pour découvrir l'IP de votre container :

```
docker inspect tpTwitter
```

Ainsi, vous pourrez joindre en ssh votre container docker. Pour cela ouvrir un nouvel onglet sur votre terminal (Ctrl + T) et lancer ssh root@IPContainer, password : 'pwd'

Pour lancer le serveur MongoDB en démon dans votre container Docker :

```
mongod --dbpath /home/db_mongo --noprealloc --nojournal --replSet testset
```

Puis connectez vous à MongoDB en lançant un client grace à la commande **mongo** dans le terminal de votre container.

Dans le client Mongo, tapez **rs.initiate()** pour activer la réplication. Ainsi les données chargées dans MongoDB pourront être répliqués sur votre serveur ElasticSearch.

Maintenant, lancer les commandes pour importer les données Twitter :

```
mongoimport -d infra_prod -c tweets_sentiment --fieldFile /home/db_files/fieldFile
--type csv --file /home/db_files/20000pos.csv
mongoimport -d infra_prod -c tweets_sentiment --fieldFile /home/db_files/fieldFile
--type csv --file /home/db_files/30000neg.csv
```

Dans le container Docker, vous rendre dans /home/elasticsearch/bin et lancer le serveur ElaSticSearch comme un démon :

```
./elasticsearch
```

Dans un autre terminal connecté en ssh au container docker, lancer les commandes suivantes :

```
curl -XPUT "localhost:9200/_river/tw_sentiment/_meta" -d '
{
"type": "mongodb",
"mongodb": {
"servers": [
{ "host": "127.0.0.1", "port": 27017 }
],
"options": { "secondary_read_preference": true },
"db": "infra_prod",
"collection": "tweets_sentiment"
},
"index": {
"name": "tw_sentiment",
"type": "tweet"
}
}'
 curl 'localhost:9200/_cat/indices?v'
```

Puis vérifier que les documents ont été indexés dans ElasticSearch :

```
curl 'localhost:9200/_cat/indices?v'
```

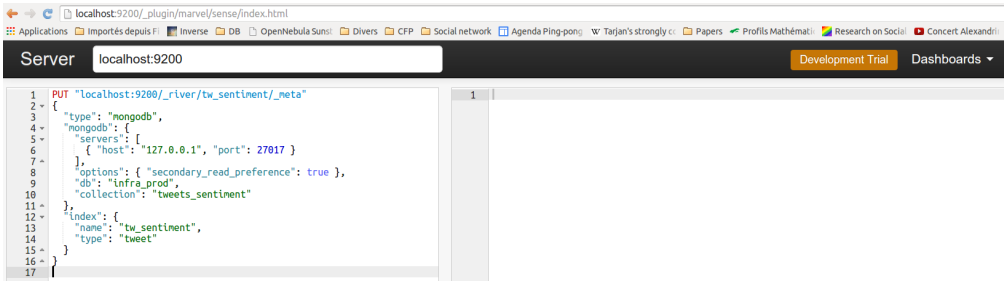
Télécharger sur votre machine Kibana : http://www.elasticsearch.org/overview/kibana/installation/

Kibana peut etre utilisé en local, le port dédié à ElasticSearch dans votre container est redirigé sur votre machine en local. Configurez donc Kibana en précisant qu'ElasticSearch tourne sur localhost:9200.

Rappel

Pour lire le JSON retourné par ElasticSearch, vous pouvez utiliser http://json.parser.online.fr/.

Avec Marvel/Sense :



Quelques Requêtes :

Avec MongoDB, obtenir les dates minimum et maximum du jeu de données.

Avec MongoDB et ElasticSearch, obtenir les tweets postés par "rhidown".

Formulez les requêtes MongoDB et ElasticSearch capables de retourner les 10 utilisateurs ayant postés le plus postés de tweets positifs ou négatifs -> http://docs.mongodb.org/manual/tutorial/aggregation-zip-code-data-set/ http://www.elasticsearch.org/guide/en/elasticsearch/reference/current/_executing_aggregations.html

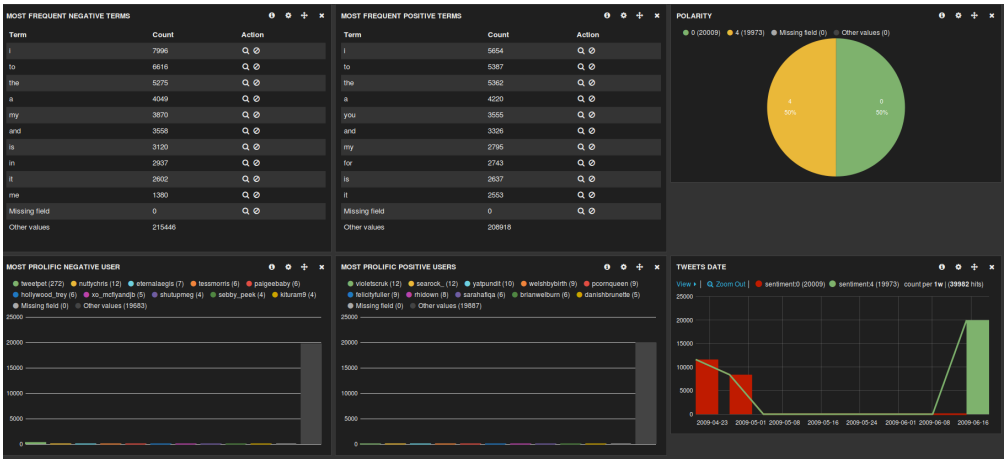
Kibana : un Dashboard pertinent

L'objectif sera de restituer visuellement à travers des panels graphiques un résumé des données et les statistiques qui nous intéressent.

Sauvegardez votre Dashboard régulièrement avec la disquette en haut à droite, il doit être exporté à la fin du TP.

On souhaite ainsi être capable de :

- Requêter les tweets de polarité négative/positive : http://www.elasticsearch.org/guide/en/kibana/current/working-with-queries-and-filters.html ;
- Afficher sous forme de tableau les termes les plus fréquents dans les tweets positifs/négatifs du jeu de données : http://www.elasticsearch.org/guide/en/kibana/current/panels.html - Utiliser un terms panel
- Afficher sous forme d'histogramme les 10 utilisateurs ayant postés le plus postés de tweets positifs/négatifs (terms panel)
- Afficher un histogramme résumant les dates auxquels les tweets ont été postés (histogram panel)



Les données

Gestion de la Date

Pour consulter le type des objets stockés dans ElasticSearch :

```
GET /tw_sentiment/_mapping
```

Par défaut, le field date qui correspond au champ mongoDB est de type string. Il est donc impossible de l'utiliser pour tracer un histogramme en fonction du temps sur Kibana.

Pour obtenir des données de type date exploitable, mettre à jour la collection MongoDB en ajoutant un nouveau champ date_new de type date à tous les documents de la database. Pour cela, vous pouvez vous aider du code ci-dessous qui montre comment convertir une string en date et des **Cursor** MongoDB pour boucler sur tous les documents et les mettre à jour :

```
testset:PRIMARY> var test = db.tweets_sentiment.findOne()
testset:PRIMARY> test.date
Mon Apr 06 22:20:03 PDT 2009
testset:PRIMARY> new Date(test.date)
ISODate("2009-04-07T05:20:03Z")
```

On obtient un nouveau mapping qui permet de dessiner l'histogramme des dates dans Kibana :

```
{
   "tw_sentiment": {
      "mappings": {
         "tweet": {
            "properties": {
               "date": {
                  "type": "string"
               },
               "date_new": {
                  "type": "date",
                  "format": "dateOptionalTime"
               },
               "query": {
                  "type": "string"
               },
               "sentiment": {
                  "type": "long"
               },
               "tweet": {
                  "type": "string"
               },
               "tweet_id": {
                  "type": "long"
               },
               "user": {
                  "type": "string"
               }
            }
         }
      }
   }
}
```

Un dataset équilibré

Avec MongoDB, faire en sorte, en supprimant des documents, d'avoir un jeu de données équilibré avec 50% de tweets étiquetés positifs, 50% de tweets étiquetés négatifs.

Dans la plupart des cas, avoir un jeu de données équilibré simplifie l'apprentissage du modèle destiné à analyser de futurs tweets.

Facultatif :

On constate que les mots les plus fréquents ne sont pas pertinents pour classifier les tweets, ils ne sont pas caractéristiques d'une polarité positive ou négative. Ce sont des mots anglais fréquents, peu importe la polarité de la phrase.

La plupart de ces mots sont appelés stopwords, il s'agit de les supprimer dans MongoDB pour améliorer la pertinence des informations restituées par le Dashboard Kibana.

Pour supprimer ces mots dans MongoDB, utilisez la liste de mots : englishStopWords

Par ailleurs, remplacer dans les fields "tweet" en utilisant des regex :

- les url par le mot clé URL;
- les mentions "@user_name" par le mot-clé USERNAME;
- Les répétitions de n lettres par des répétitions de 2 lettres seulement : les "huuuuuuge" par "huuge", les "goooo" par "goo".

Toutes ces opérations sur les données accéléreront la recherche et amélioreront la pertinence des futurs classifieurs.

Quelques requêtes MapReduce

- Compter le nombre d'occurences de chaque mot dans le jeu de données
- Compter le nombre de tweets posté par chaque utilisateur

- Compter le nombre de caractères moyen par tweets pour chaque polarité

Un premier classifieur naïf

En utilisant deux listes de mots anglais positifs et négatifs (http://www.enchantedlearning.com/wordlist/positivewords.shtml http://www.enchantedlearning.com/wordlist/negativewords.shtml), on souhaite créer un classifieur basique.

Ce classifieur va être implémenté en utilisant l'API ElasticSearch et notamment grace aux puissants Analyzers.

Les analyzers permettent de préciser une méthode pour indexer les documents et analyser des requêtes de recherche.

Exécutez par exemple le code suivant pour constater les effets d'un Analyzer capable de détecter des mots comme négatifs :

```
POST /classifieur_negs_stopwords/_close
PUT /classifieur_negs_stopwords
{
    "settings" : {
        "analysis" : {
            "char_filter" : {
                "negs" : {
                    "type" : "pattern_replace",
                    "pattern" : "\\b((?i:never|no)\\b",
                    "replacement" : "NEG"
                }
            },

            "analyzer" : {
                "negcon_tagger" : {
                    "type" : "custom",
                    "tokenizer" : "whitespace",
                    "filter" : ["lowercase", "kstem"],
                    "char_filter" : ["negs"]
                }
            }
        }
    }
}
POST /classifieur_negs_stopwords/_open
GET /classifieur_negs_stopwords/_analyze?analyzer=negcon_tagger&text=The boy was
not happy. He is never happy. Blablabla.
```

Exécutez ce second exemple pour constater qu'ElasticSearch est capable de retirer lui meme les stopwords de ses requêtes :

```
PUT /english_docs
{
    "settings": {
        "analysis": {
            "analyzer": {
                "es_std": {
                    "type":      "standard",
                    "stopwords": "_english_"
                }
            }
        }
    }
}
GET /english_docs/_analyze?analyzer=es_std&text=The boy was not happy. His friend
bangged his mother. Too bad...
```

Il s'agit donc maintenant de créer un Analyzer capable de supprimer les stopwords des requêtes et de remplacer les mots à connotations négatives par NEG, celles à connotations positifs par POS.

Vous modifierez ensuite le mapping de votre index en ajoutant un champ tweet_new dont l'analyzer sera celui que vous avez créé.

Dans MongoDB, vous copierez pour toutes les entrées le contenu du field tweet dans un nouveau field tweet_new. Celui-ci sera ainsi indexé en utilisant votre nouvel Analyzer.

Ainsi, une fois ces taches effectuées, en recherchant des documents de la façon qui suit, il vous sera possible d'obtenir les documents à polarité négatives :

```
GET /tw_sentiment/_search?q=tweet_new:NEG
```

Doc : http://www.elasticsearch.org/guide/en/elasticsearch/reference/current/analysis-custom-analyzer.html

http://www.elasticsearch.org/guide/en/elasticsearch/guide/current/configuring-analyzers.html

http://www.elasticsearch.org/guide/en/elasticsearch/guide/current/mapping-intro.html

http://www.elasticsearch.org/guide/en/elasticsearch/reference/current/indices-analyze.html