Indexing and search plugins for Elasticsearch 1.3.2

SemanticAnalyzer

www.semanticanalyzer.info

29 July, 2015

This document takes you through the steps necessary to install the plugins for indexing and searching your content in the Russian language. Both formal texts and informal short texts (tweets) are supported. The plugins are based on the linguistic components of SemanticAnalyzer. The following components have been implemented for Elasticsearch:

- Tokenizer that supports tokenizing formal and informal texts
- Lemmatizing filter based on the lemmatizer for the Russian language by SemanticAnalyzer

Installation

In order to install the plugins, navigate to Elasticsearch HOME_DIR and issue a command:

> bin/plugin -install analysis-morphology -url file:/home/dmitry/projects/elasticsearch/elasticsearch-analysis-morphology/target/releases/elasticsearch-analysis-morphology-1.0.zip

This will output to the console:

```
-> Installing analysis-morphology...

Trying file:/home/dmitry/projects/elasticsearch/elasticsearch-analysis-
morphology/target/releases/elasticsearch-analysis-morphology-1.0.zip...

Downloading ....DONE

Installed analysis-morphology into
/home/dmitry/projects/elasticsearch/elasticsearch-1.2.1/plugins/analysis-
morphology
```

For convenience, the delivery package contains the update_plugin.sh script that will perform all the necessary actions, such as removing the old version of the plugin (in case of reinstallation) and installing the new version.

Next, startup the Elasticsearch server, using:

> bin/elasticsearch

The output should be along the lines:

```
[2015-07-28  18:06:30,514][INFO  ][node                                    ] [Valkyrie]
version[1.2.1], pid[2816], build[6c95b75/2014-06-03T15:02:52Z]
```

```
[2015-07-28 18:06:30,524][INFO ][node                                ] [Valkyrie] initializing ...
[2014-07-28 18:06:30,584][INFO ][plugins                             ] [Valkyrie] loaded [analysis-morphology], sites []
```

Notice the line in **bold**: it shows, that analysis-morphology plugin has been loaded. This means, the plugin installation went successfully.

Running tests

The delivery package includes two scripts that run real tests against the Elasticsearch and plugins you installed in the previous section. They are:

demo_generic_tokenizer.sh

demo_twitter_tokenizer.sh

The difference between these two is that:

the first script uses generic tokenizer to handle formal texts

the second script uses twitter tokenizer to handle twitter texts

Run the first script as follows:

> ./demo_generic_tokenizer.sh

You should see the following output:

```
{"acknowledged":true}
{"acknowledged":true}
{"acknowledged":true}
{"_index":"rustest","_type":"type1","_id":"1","_version":1,"created":true}
{"_index":"rustest","_type":"type1","_id":"2","_version":1,"created":true}
{"_index":"rustest","_type":"type1","_id":"3","_version":1,"created":true}
{"_index":"rustest","_type":"type1","_id":"4","_version":1,"created":true}
{"_index":"rustest","_type":"type1","_id":"5","_version":1,"created":true}
{"_index":"rustest","_type":"type1","_id":"6","_version":1,"created":true}
{"_shards":{"total":10,"successful":5,"failed":0}}
Should return 5
      "_id" : "5",
Should return 4, 6
```

```
      "_id" : "6",
      "_id" : "4",
Should return 4
      "_id" : "4",
Should return 1, 4
      "_id" : "1",
      "_id" : "4",
Should return 2
      "_id" : "2",
Should return 3
      "_id" : "3",
Should return 1,2,3,4
      "_id" : "4",
      "_id" : "1",
      "_id" : "2",
      "_id" : "3",
```

Run the first script as follows:

> ./demo_twitter_tokenizer.sh

You should see the following output:

```
{"acknowledged":true}
{"acknowledged":true}
{"acknowledged":true}
{"_index":"rustest","_type":"type1","_id":"1","_version":1,"created":true}
{"_index":"rustest","_type":"type1","_id":"2","_version":1,"created":true}
{"_index":"rustest","_type":"type1","_id":"3","_version":1,"created":true}
{"_index":"rustest","_type":"type1","_id":"4","_version":1,"created":true}
{"_index":"rustest","_type":"type1","_id":"5","_version":1,"created":true}
{"_index":"rustest","_type":"type1","_id":"6","_version":1,"created":true}
{"_index":"rustest","_type":"type1","_id":"7","_version":1,"created":true}
{"_shards":{"total":10,"successful":5,"failed":0}}
Should return 5
      "_id" : "5",
Should return 4, 6
      "_id" : "6",
      "_id" : "4",
```

```
Should return 4
      "_id" : "4",
Should return 1, 4
      "_id" : "1",
      "_id" : "4",
Should return 2
      "_id" : "2",
Should return 3
      "_id" : "3",
Should return 1,2,3,4
      "_id" : "4",
      "_id" : "1",
      "_id" : "2",
      "_id" : "3",
Should return 7
      "_id" : "7",
Should return 7
      "_id" : "7",
Should return nothing
```

The scritps demonstrate how to create an index with a particular analyzer (for formal texts or tweets) and index data into the respective index. Next each script performs a number of queries and outputs both what it expects and what has been returned by Elasticsearch. So the user is able to assess whether the tests passed.