Συστήματα Ανάκτηση Πληροφοριών – Πρώτο Παραδοτέο

Ηλίας Σέττας - 3150156

Η επικοινωνία με την elasticsearch πραγματοποιείται με τη χρήση python client και βιβλιοθήκης elasticsearch.

ΣΗΜΑΝΤΙΚΑ:

1) Σε περίπτωση που θέλετε να τρέξετε ολόκληρο το πρόγραμμα και να ανεβάσετε με αυτό τα αρχεία στο Elastic Search τότε πρέπει να τοποθετήσετε το αρχείο Parsed files μέσα στο φάκελο του παραδοτέου.

2) Ανοίξτε το project σε PyCharm και εγκαταστήστε τις αναγκαίες βιβλιοθήκες για το πρόγραμμα (elasticsearch και xmltodict).

3) Εκκινήστε την Elastic Search και τρέξτε το πρόγραμμα (διαβάστε οδηγίες παρακάτω).

4) Όλα τα υπόλοιπα απαραίτητα (trec_eval, qrels.txt) και τα αποτελέσματα βρίσκονται ήδη στο φάκελο του project.

Η λειτουργία του client ρυθμίζεται από τη μεταβλητή mode. Αν θέλετε να πραγματοποιήσετε εισαγωγή δεδομένων από το Parsed files προς την elasticsearch τότε θέστε τη μεταβλητή ως 'all'. Αν θέλετε απλώς ο client να κάνει αναζήτηση μόνο για τα 20 και 30 πιο σχετικά κείμενα με βάση τις ερωτήσεις θέστε τη ως 'other'. Σε αυτή την περίπτωση θα παραλείψει το πρώτο κομμάτι και θα στείλει μόνο αιτήματα στο server για αναζήτηση.

Το αποτέλεσμα θα είναι 2 αρχεία (es_results_20.txt και es_results_30.txt), τα οποία θα περιέχουν τις πληροφορίες σχετικά με τις ανακτήσεις που γίνανε. Στη συνέχεια με τη βοήθεια του trec_eval γίνεται η σύγκριση με το αρχείο qrels.txt και

δημιουργούνται άλλα 2 νέα αρχεία αποτελεσμάτων (eval_results_20.txt και eval_results_30.txt). Όπως αναφέρθηκε και πριν τα αποτελέσματα αυτά υπάρχουν ήδη στο φάκελο από προηγούμενες εκτελέσεις του κώδικα.

Οι 2 φάσεις του προγράμματος:

1) Ο client διαβάζει όλα τα xml αρχεία από το φάκελο Parsed files και με τη χρήση της βιβλιοθήκης xmltodict τα μετατρέπει όλα σε dictionaries αποθηκεύοντάς στα στη μεταβλητή doc. Στη συνέχεια, για κάθε dictionary από το doc, δημιουργείται ένα καινούργιο στο οποίο τα πεδία objective + title είναι ενωμένα σε ένα πεδίο με όνομα text και αποθηκεύεται στη μεταβλητή package. Με τη χρήση της βιβλιοθήκης elasticsearch τα συγκεκριμένα κάθε package - dictionary ανεβαίνει στην βάση δεδομένων που έχουμε ανοιχτή. Φυσικά, πριν γίνει αυτή η διαδικασία ο client στέλνει ένα dictionary - body με τις απαραίτητες ρυθμίσεις σχετικά με analyzer και filters (english).

```
body={
    'mappings': {
        'project': {
            'properties': {
                'rcn': {
                    'type': 'integer'
                },
                'acronym': {
                    'type': 'string'
                },
                'text': {
                    'type': 'string',
                    'analyzer': 'english',
                    'search_analyzer': 'english'
                },
                'identifier': {
                    'type': 'string'
                }
            }
        }
    }
}
```

Μετά από αυτή τη διαδικασία στέλνεται και ένα τελευταίο dictionary που περιέχει τις ρυθμίσεις σχετικά με τα βάρη (TF-IDF).

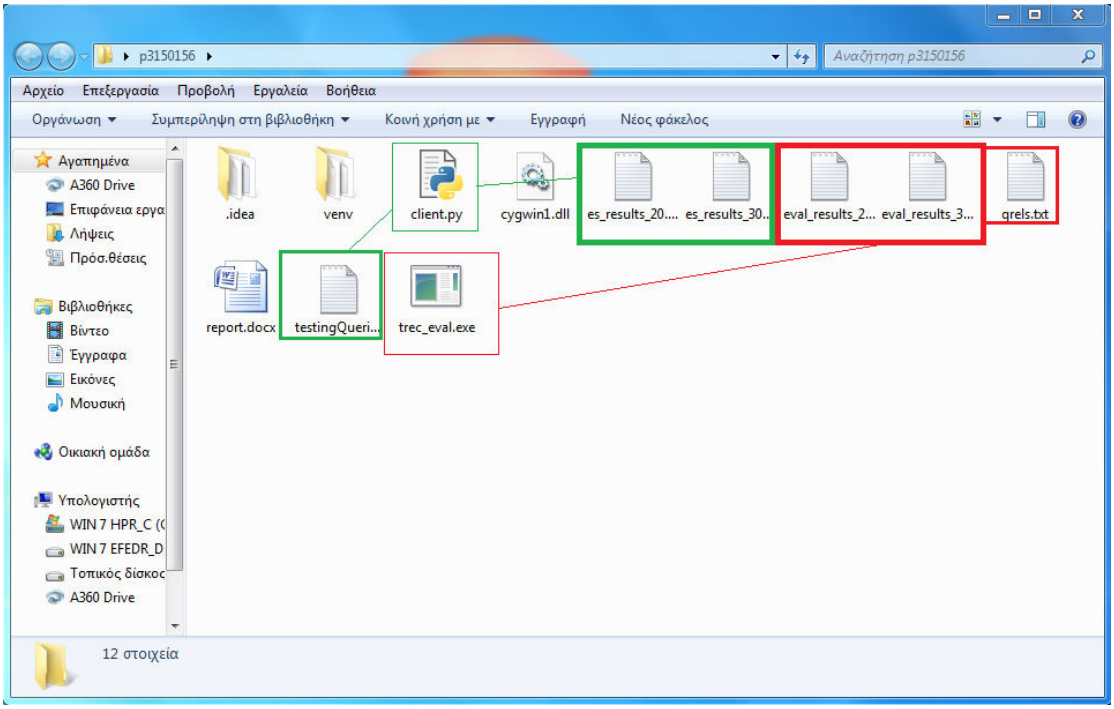
```
es.indices.close(index='test')
es.indices.put_settings(
    index='test',
    body={
        'index': {
            'similarity': {
                'default': {
                  'type': 'classic'
                }
            }
        }
    }
)
es.indices.open(index='test')
time.sleep(1)
```

Και κάπως έτσι ολοκληρώνεται η διαδικασία τροποποίησης και ανεβάσματος των xml αρχείων. Στο αρχείο python που έχει παραδοθεί η μεταβλητή έχει τεθεί στο ‘other’ ώστε να παραλείπεται αυτή η διαδικασία και να εκτελείται μόνο το δεύτερο κομμάτι.

2) Ο client διαβάζει το αρχείο με τις ερωτήσεις γραμμή προς γραμμή αφαιρώντας τα αρχικά tags (Q#) και κρατώντας το κείμενο (text) στη μεταβλητή line. Για κάθε γραμμή στέλνει 2 αιτήματα για search - αναζήτηση στην Elastic Search, ένα για 21 πιο σχετικά και ένα για 31 πιο σχετικά κείμενα. Από τις απαντήσεις αφαιρείται το πρώτο πιο σχετικό κείμενο και κρατάμε τα υπόλοιπα. Τέλος γράφει τα αποτελέσματα στα αρχεία es_results_20.txt και es_results_30.txt.

```
#We need k+1 results for each test
result20 = es.search(index='test', doc_type='project', body={'query': {'match': {'text': line}}, 'size': 21})
result30 = es.search(index='test', doc_type='project', body={'query': {'match': {'text': line}}, 'size': 31})
```

Τα 2 αυτά αρχεία σε συνδυασμό με το αρχείο qrels.txt χρησιμοποιούνται από τον trec_eval για την τελική αξιολόγηση των ανακτήσεων. Τα αποτελέσματα βρίσκονται στα αρχεία eval_results_20.txt και eval_results_30.txt.




```
271  runid                   all test
272  num_q                   all 10
273  num_ret                 all 200
274  num_rel                 all 142
275  num_rel_ret             all 91
276  map                     all 0.4720
277  gm_map                  all 0.4436
278  Rprec                   all 0.5459
279  bpref                   all 0.6259
280  recip_rank              all 0.9500
281  iprec_at_recall_0.00    all 0.9857
282  iprec_at_recall_0.10    all 0.9407
283  iprec_at_recall_0.20    all 0.7932
284  iprec_at_recall_0.30    all 0.7377
285  iprec_at_recall_0.40    all 0.6222
286  iprec_at_recall_0.50    all 0.5303
287  iprec_at_recall_0.60    all 0.4373
288  iprec_at_recall_0.70    all 0.2765
289  iprec_at_recall_0.80    all 0.0688
290  iprec_at_recall_0.90    all 0.0000
291  iprec_at_recall_1.00    all 0.0000
292  P_5                     all 0.7000
293  P_10                    all 0.5900
294  P_15                    all 0.5467
295  P_20                    all 0.4550
296  P_30                    all 0.3033
297  P_100                   all 0.0910
298  P_200                   all 0.0455
299  P_500                   all 0.0182
300  P_1000                  all 0.0091
```