Ερωτήματα Κειμένων (QuerybyDocument)

Για την εργασία θα χωριστείτε σε δύο ομάδες: Στην ομάδα 1 (Ο1) θα ανήκουν αυτοί των οποίων ο αριθμός μητρώου λήγει σε περιττό αριθμό και στην ομάδα 2 (Ο2) αυτοί των οποίων ο αριθμός μητρώου λήγει σε άρτιο αριθμό (0,2,4..)

Περισσότερες λεπτομέρειες για την εργασία θα δίνονται στις διαλέξεις του μαθήματος.

Δίνεται μια συλλογή XMLκειμένων.Τα κείμενα περιέχουν τα εξής πεδία:

<rcn> (recordnumber) : το μοναδικό αναγνωριστικό (ID) του κειμένου
<acronym> : Μια συμβολοσειρά που παριστά το ακρώνυμο του κειμένου.
<title>: Ο τίτλος του κειμένου
<objectives>: Η περίληψη του κειμένου
<identifier>: μια συμβολοσειρά που δηλώνει την κατηγορία του κειμένου

ΦΑΣΗ 1

Με βάση τη συλλογή και 10 κείμενα που θα σας δοθούν ως ερωτήματα θα χρησιμοποιήσετε τη μηχανή αναζήτησης ElasticSearchκαι θα ανακτήσετε τα κπλησιέστερα κείμενα. Επειδή τα ερωτήματα είναι κείμενα της συλλογής το πρώτο κείμενο που θα ανακτήσουν είναι ο εαυτός τους. Θα πρέπει να εξαιρέσετε από την απάντηση το κείμενο αυτό και να επιστρέψετε τα κείμενα 2…k+1. Δοκιμάστε τιμές k=20, 30.

Αξιολογείστε τα αποτελέσματά σας με το trec_eval και τα μέτρα αξιολόγησης, MAP, (Mean Average Precision)και avgPre@k, (μέση ακρίβεια στα κπρώτα ανακτηθέντα κείμενα) με k=5, 10, 15, 20).

Η ομάδα Ο1 θα χρησιμοποιήσει βάρη BM25 και η Ο2 βάρη TFIDF.

Παράδοση 14 Απριλίου 2019

(Δείτε μερικές διευκρινήσεις για τη φάση 1 σε ξεχωριστό αρχείο στο ίδιο φάκελο)

ΦΑΣΗ 2Α

Ομάδα Ο1: Χρησιμοποιώντας το API της Yahoo[1] – term extraction θα εξαγάγετε για κάθε κείμενο λέξεις κλειδιά και φράσεις από τα πεδία <title>και <objectives>.

2Β

Ομάδα Ο2:Όμοια με το 2Α αλλά θα χρησιμοποιήσετε το API της Microsoft[2]για την εξαγωγή λέξεων και φράσεων από τα κείμενα.

2Γ

(**Όποιος χρησιμοποιήσει αυτή την επιλογή θα πρέπει να ενημερώσει τους διδάσκοντες**) Όμοια με παραπάνω αλλά θα χρησιμοποιήσετε ένα άλλο API ή εργαλείο (αιτιολογώντας γιατί το επιλέξατε) που θα βρείτε στο διαδίκτυο.

2Δ ++

(**Όποιος χρησιμοποιήσει αυτή την επιλογή θα πρέπει να ενημερώσει τους διδάσκοντες**) Χρησιμοποιείστε τη εφαρμογή WikipediaMiner[3] [1]και εξάγετε από όλα τα κείμενα concepts (έννοιες-λήμματα) από την Wikipedia.

Το αποτέλεσμα της φάσης αυτής θα είναι μια νέα έκδοση της συλλογής (συλλογή-2)η οποία θα περιλαμβάνει:

[1]https://www.programmableweb.com/api/yahoo-term-extraction
[2]https://azure.microsoft.com/en-us/services/cognitive-services/text-analytics/
[3]https://github.com/dnmilne/wikipediaminer/wiki/Wikipedia-Miner

<**rcn**> : το μοναδικό αναγνωριστικό (ID) του κειμένου
<**acronym**> : Μια συμβολοσειρά που παριστά το ακρώνυμο του κειμένου.
<**keywords**>: οι όροι που εξήχθησαν από τα πεδία <title> και <objectives>.

Παράδοση 10 Μαΐου 2019

ΦΑΣΗ 3 **(και οι δύο ομάδες)** Με βάση τη συλλογή-2 και τα νέα ερωτήματα (που περιέχουν μόνο τις φράσεις που εξήχθησαν από τα αρχικά κείμενα) θα χρησιμοποιήσετε το ElasticSearchκαι θα επαναλάβετε τη ΦΑΣΗ 1δηλαδή θα ανακτήσετε τα κπλησιέστερα κείμενα από τη συλλογή-2. Δοκιμάστε να αναπαραστήσετε τα κείμενα με διαφορετικά ποσοστά των λέξεων κλειδιών που περιέχουν, π.χ. δοκιμάστε τα 20% καλύτερα keywords-φράσεις, 30%, 40%, 50% κλπ.

Αξιολογείστε τα αποτελέσματά σας με το trec_eval (avgPre@k, k=5, 10, 20)

ΦΑΣΗ 4 **(και οι δύο ομάδες: η Ο1 θα χρησιμοποιήσει το Sim_{BM25}και η Ο2 το Sim_{TFIDF})** Αν Sim_{BM25} (q,d) δηλώνει την ομοιότητα μεταξύ του ερωτήματος και κειμένου όπως προέκυψε από τη ΦΑΣΗ 1 και sim_{phrase}(q,d) την ομοιότητα του ερωτήματος και κειμένου όπως προέκυψε από τη ΦΑΣΗ 3 τότε επαναταξινομήστε τα κείμενα που ανακτώνται από κάθε ερώτημα με βάση τη σχέση:

$$SIM(q,d) = \lambda \cdot sim_{BM25}(q,d) + (1-\lambda) \cdot sim_{phrase}(q,d) \qquad (1)$$

Δοκιμάστε διάφορες τιμές της παραμέτρου λ έτσι ώστε η επίδοση της (1) να είναι καλύτερη από την επίδοση και των δύο συναρτήσεων Sim_{BM25} (q,d) και sim_{phrase}(q,d).

Παράδοση 1 Ιουλίου 2019

[1] An open-source toolkit for mining Wikipedia, David Milne,IanH.Witten, Artificial Intelligence 194, 222-239, 2013

The online encyclopedia Wikipedia is a vast, constantly evolving tapestry of interlinked articles. For developers and researchers it represents a giant multilingual database of concepts and semantic relations, a potential resource for natural language processing and many other research areas. This paper introduces the Wikipedia Miner toolkit, an open-source software system that allows researchers and developers to integrate Wikipedia's rich semantics into their own applications. The toolkit creates databases that contain summarized versions of Wikipedia's content and structure, and includes a Java API to provide access to them. Wikipedia's articles, categories and redirects are represented as classes, and can be efficiently searched, browsed, and iterated over. Advanced features include parallelized processing of Wikipedia dumps, machine-learned semantic relatedness measures and annotation features, and XML-based web services. Wikipedia Miner is intended to be a platform for sharing data mining techniques.