

ΜΗΧΑΝΗ ΑΝΑΖΗΤΗΣΗΣ «GOOGLE»

ΜΕ ΧΡΗΣΗ ΤΗΣ ΒΙΒΛΙΟΘΗΚΗΣ LUCENE
ΚΑΙ ΤΗΣ ΓΛΩΣΣΑΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ JAVA
ΣΕ ΠΕΡΙΒΑΛΛΩΝ ECLIPSE ΤΟΥ WINDOWS.

Ταφλαμπας Νικολαος 4500

Juulia Jasmiina Jyrhä 2665

ΜΥΕ003: Ανάκτηση Πληροφορίας Εαρινό Εξάμηνο 2020-2021

ΠΕΡΙΕΧΟΜΕΝΑ

ΦΑΣΗ I

| | |
|--|---|
| Κατέβασμα βιβλιοθηκών – στήσιμο προγράμματος ----- | 3 |
| Στόχος και λειτουργικότητα ----- | 4 |
| Συλλογή εγγράφων (corpus) ----- | 5 |
| Full JSON schema ----- | 6 |
| Δημιουργία των Document ----- | 8 |
| Η αναζήτηση ----- | 8 |
| Αρχικό GUI & Παρουσίαση αποτελεσμάτων ----- | 9 |

ΦΑΣΗ II

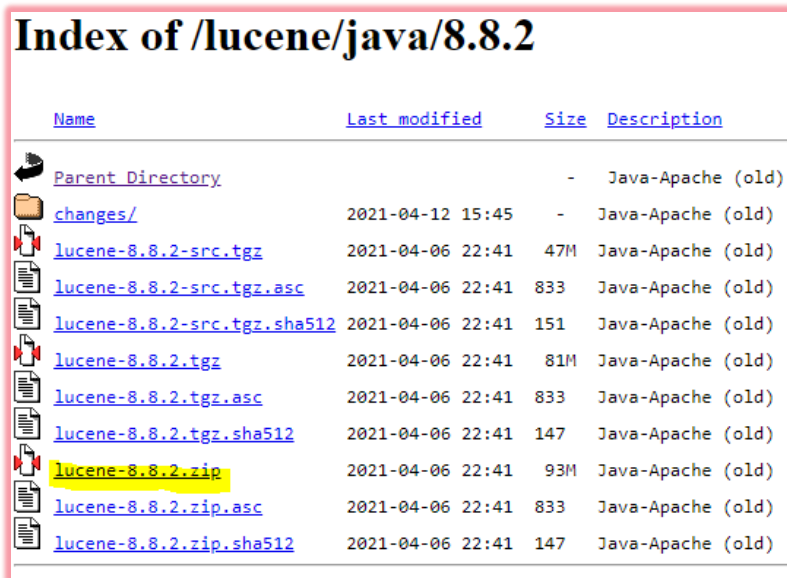
| | |
|-----------------------------------|----|
| Η τελική γραφική διεπαφή ----- | 10 |
| Το Set-Up της εφαρμογής μας ----- | 10 |
| Αναζήτηση με το GOGGLE ----- | 12 |
| Απόκτηση του GOGGLE & DEMO ----- | 13 |

ΠΡΩΤΗ ΦΑΣΗ ΤΟΥ PROJECT

ΚΑΤΕΒΑΣΜΑ ΤΩΝ ΒΙΒΛΙΟΘΗΚΩΝ – ΣΤΗΣΙΜΟ ΠΡΟΓΡΑΜΜΑΤΟΣ

Για να δημιουργήσουμε ένα Project με χρήση της βιβλιοθήκης Lucene στο περιβάλλον JavaEclipse των WINDOWS πρέπει να γίνουν οι εξής ενέργειες:

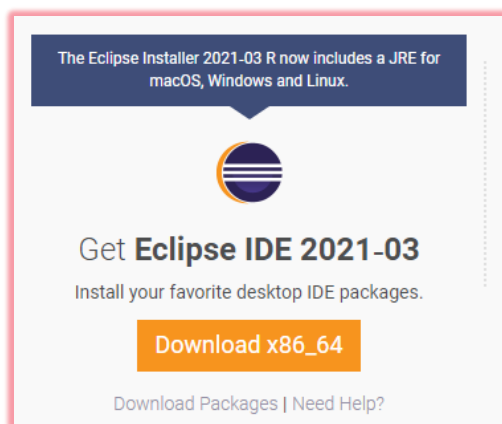
- ΚΑΤΕΒΑΣΜΑ ΤΗΣ LUCENE μέσω του συνδέσμου: <https://downloads.apache.org/lucene/java/8.8.2/> (ή κάποιας έκδοσης της lucene για Java)
Όπου επιλέγουμε το αρχείο .zip.



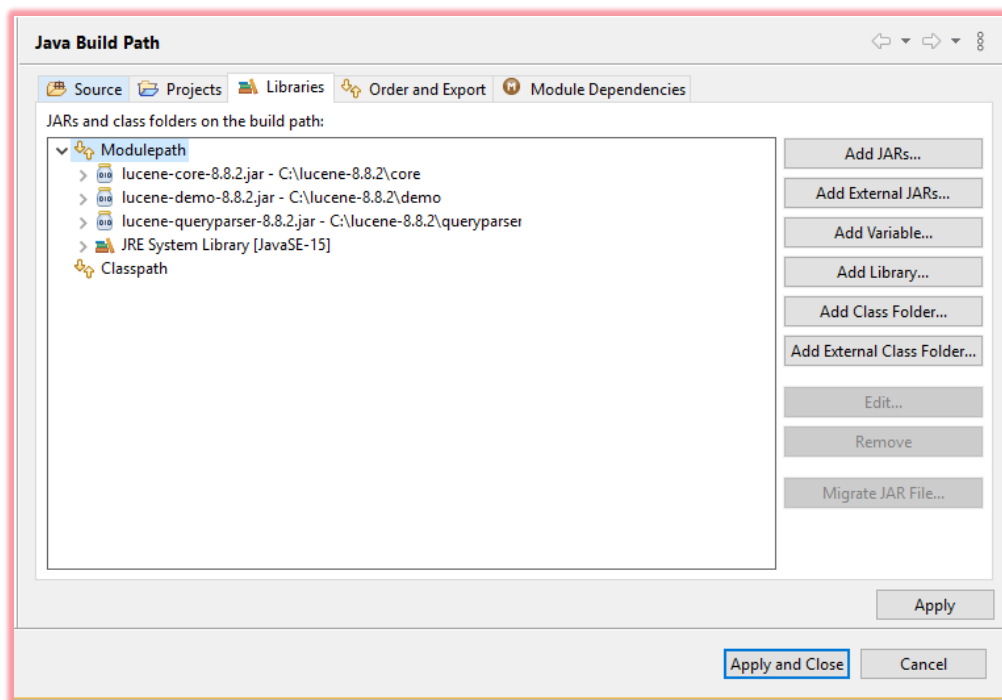
| Name | Last modified | Size | Description |
|---|------------------|------|-------------------|
| Parent Directory | | - | Java-Apache (old) |
| changes/ | 2021-04-12 15:45 | - | Java-Apache (old) |
| lucene-8.8.2-src.tgz | 2021-04-06 22:41 | 47M | Java-Apache (old) |
| lucene-8.8.2-src.tgz.asc | 2021-04-06 22:41 | 833 | Java-Apache (old) |
| lucene-8.8.2-src.tgz.sha512 | 2021-04-06 22:41 | 151 | Java-Apache (old) |
| lucene-8.8.2.tgz | 2021-04-06 22:41 | 81M | Java-Apache (old) |
| lucene-8.8.2.tgz.asc | 2021-04-06 22:41 | 833 | Java-Apache (old) |
| lucene-8.8.2.tgz.sha512 | 2021-04-06 22:41 | 147 | Java-Apache (old) |
| lucene-8.8.2.zip | 2021-04-06 22:41 | 93M | Java-Apache (old) |
| lucene-8.8.2.zip.asc | 2021-04-06 22:41 | 833 | Java-Apache (old) |
| lucene-8.8.2.zip.sha512 | 2021-04-06 22:41 | 147 | Java-Apache (old) |

Αφού το κατεβάσουμε, το αποσυμπιέζουμε (με χρήση ενός προγράμματος αποσυμπίεσης πχ 7Zip) .
Έπειτα μετακινούμε τον φάκελο που δημιουργήθηκε, έστω στο C:\lucene-8.8.2

- ΣΤΗΣΙΜΟ ECLIPSE από τον σύνδεσμο: <https://www.eclipse.org/downloads/>



- ΕΙΣΑΓΩΓΗ ΤΗΣ LUCIND ΣΤΟ ECLIPSE
Αυτό μπορούμε να το κάνουμε δημιουργώντας ένα project στο Eclipse και κάνοντας δεξί κλικ και επιλέγοντας Properties > Java Build Path > Libraries > Modulepath > Add External JARs



Τα απαραίτητα JAR αρχεία φαίνονται στην παραπάνω εικόνα, όπως και τα directories μέσα στα οποία βρίσκονται. Αφού προστεθούν τα 3 αυτά τα αρχεία, κάνουμε “Apply and Close”

- ΥΠΟΣΤΗΡΙΞΗ ΑΡΧΕΙΩΝ JSON

Για να μπορούμε να επεξεργαστούμε JSON αρχεία, μεταβαίνουμε στο: <https://jar-download.com/artifacts/org.json>, πάμε στα μισά της ιστοσελίδας και βρίσκουμε το κουμπί download.



Κάνουμε unzip, και προσθέτουμε το jar αρχείο στα ModulePath όπως κάναμε και παραπάνω.

ΣΤΟΧΟΣ ΚΑΙ ΛΕΙΤΟΥΡΓΗΚΟΤΗΤΑ

Ο στόχος της εργασίας είναι η υλοποίηση μιας αποδοτικής μηχανής αναζήτησης αξιοποιώντας την βιβλιοθήκη Apache Lucene. Ο κώδικας μας πρέπει να επεξεργάζεται JSON αρχεία, να δημιουργεί μια βιβλιοθήκη αρχείων αναγνώσιμα από το βασικά μας πρόγραμμα, να μπορεί να δημιουργεί Index μέσω αυτών των απλών αρχείων, και φυσικά να βρίσκει τα πιο σχετικά άρθρα βάση την λέξη/φράση αναζήτησης του χρήστη. Τέλος ένα σύστημα γραφικών GUI επίσης θα πρέπει να υλοποιηθεί με σκοπό να είναι φιλικό προς τον χρήστη.

ΣΥΛΛΟΓΗ ΕΓΓΡΑΦΩΝ (CORPUS)

Η συλλογή μας πάρθηκε από τον παρακάτω σύνδεσμο:

https://ai2-semantic scholar-cord-19.s3-us-west-2.amazonaws.com/historical_releases.html

Και είναι η πρώτη έκδοση από μια συλλογή που ονομάζεται CORD-19. Αυτή αποτελείται από ακαδημαϊκές εργασίες σχετικά με το COVID-19 και την έρευνα σχετικά με τους κοροναϊούς. Στα κείμενα έχουν ήδη πραγματοποιηθεί ορισμένοι «καθαρισμοί» των δεδομένων σύμφωνα με το readme του github της συλλογής :









<https://github.com/allenai/cord19>

Το σύνολο των δεδομένων που κατεβάσαμε εμείς είναι αυτό που δημοσιεύθηκε 13-03-2020, που ήταν επαρκές σε μέγεθος για την δημιουργία του corpus που θέλουμε να δημιουργήσουμε σε αυτή τη φάση.

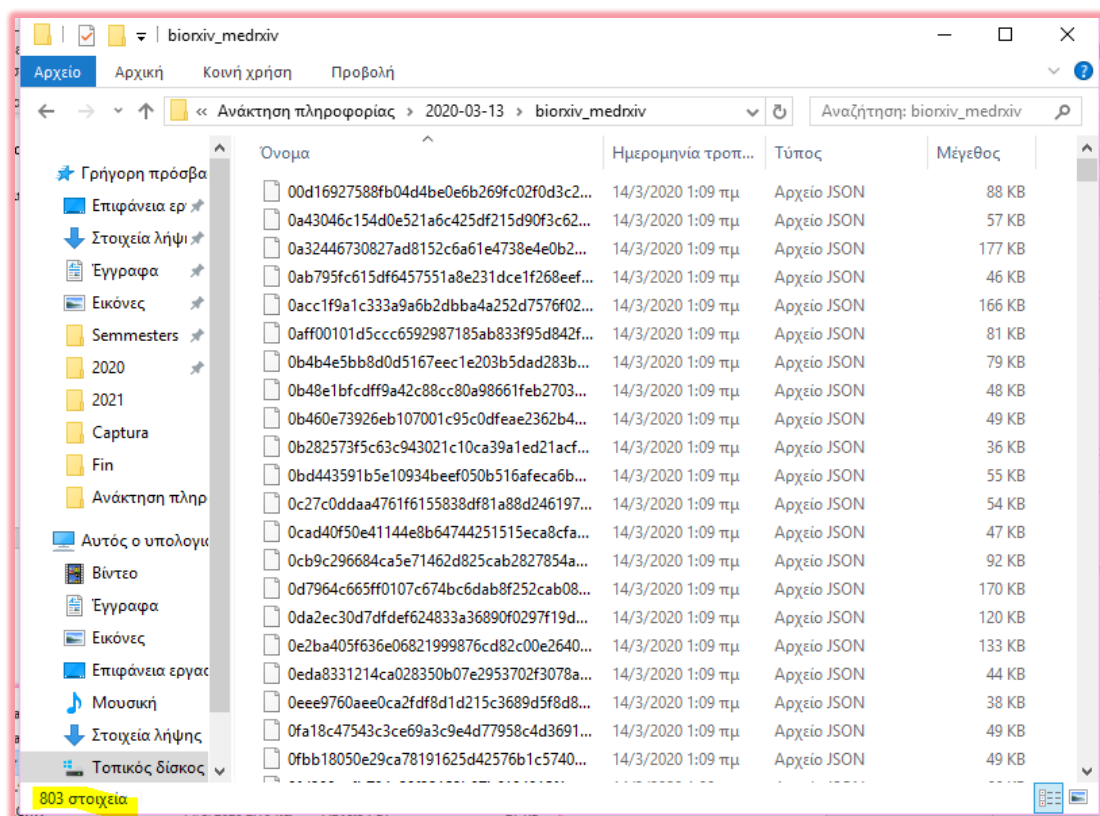
2020-03-13

cord-19_2020-03-13.tar.gz (0.3GB) a36fe181 8fbea927

Αρα: κατεβάζουμε τα αρχεία από τον πρώτο σύνδεσμο με την παραπάνω ένδειξη, τα κάνουμε unzip δύο (!) φορές, την πρώτη για να κάνουμε το αρχείο με κατάληξη .gz σε tar, και έπειτα για να μας βγάλει τον φάκελο 2020-03-13. Μετά, κάνουμε την ίδια διαδικασία με το αρχείο biorxiv_medrxiv.tar για να βρούμε τα JSON αρχεία.

| | | | | |
|---|--|-------------------|---------------------|------------|
|  | all_sources_metadata_2020-03-13 | 16/3/2020 7:30 μμ | Αρχείο τιμών δια... | 48.672 KB |
|  | all_sources_metadata_2020-03-13.readme | 6/4/2020 9:34 μμ | Αρχείο README | 2 KB |
|  | biorxiv_medrxiv.tar | 14/3/2020 1:37 πμ | Αρχείο GZ | 12.871 KB |
|  | comm_use_subset.tar | 14/3/2020 1:40 πμ | Αρχείο GZ | 190.750 KB |
|  | COVID.DATA.LIC.AGMT | 14/3/2020 2:16 πμ | Αρχείο PDF | 27 KB |
|  | json_schema | 14/3/2020 1:53 πμ | Έγγραφο κειμένου | 3 KB |
|  | noncomm_use_subset.tar | 14/3/2020 1:38 πμ | Αρχείο GZ | 36.536 KB |
|  | pmc_custom_license.tar | 14/3/2020 1:40 πμ | Αρχείο GZ | 19.375 KB |

Τα αρχεία αυτά είναι στο σύνολό τους 803.



Η απαίτηση ήταν να έχουμε μεταξύ 2.000 εγγράφων σε περίπτωση άρθρων και 500 έγγραφα στην περίπτωση σύντομων κειμένων από κοινωνικών δίκτυα, οπότε θεωρούμε ότι τα άρθρα είναι επαρκή σε αριθμό καθώς στο μέγεθος τους είναι ικανοποιητικά μεγάλα, συνήθως πάνω από 60 χιλιάδες χαρακτήρες.

JSON SCHEMA OF FULL TEXT DOCUMENTS

Η συλλογή έρχεται με ένα αρχείο json_schema που επεξηγεί την μορφή των αρχείων που περιέχονται στον φάκελο που μόλις δημιουργήσαμε παραπάνω. Το σχήμα είναι το εξής:

```
{
  "paper_id": <str>,          # 40-character sha1 of the PDF
  "metadata": {
    "title": <str>,
    "authors": [              # list of author dicts, in order
      {
        "first": <str>,
        "middle": <list of str>,
        "last": <str>,
        "suffix": <str>,
        "affiliation": <dict>,
        "email": <str>
      },
      ...
    ],
    "abstract": [             # list of paragraphs in the abstract
      {
        "text": <str>,
        "cite_spans": [       # list of character indices of inline citations
          {
            # e.g. citation "[7]" occurs at positions 151-154 in "text"
            # linked to bibliography entry BIBREF3
            "start": 151,
            "end": 154,
            "text": "[7]",
            "ref_id": "BIBREF3"
          }
        ]
      }
    ]
  }
}
```

```

    },
    ...
  ],
  "ref_spans": <list of dicts similar to cite_spans>, # e.g. inline reference to "Table 1"
  "section": "Abstract"
},
...
],
"body_text": [ # list of paragraphs in full body
  # paragraph dicts look the same as above
  {
    "text": <str>,
    "cite_spans": [],
    "ref_spans": [],
    "eq_spans": [],
    "section": "Introduction"
  },
  ...
  {
    ...,
    "section": "Conclusion"
  }
],
"bib_entries": {
  "BIBREF0": {
    "ref_id": <str>,
    "title": <str>,
    "authors": <list of dict> # same structure as earlier,
    # but without `affiliation` or `email`
    "year": <int>,
    "venue": <str>,
    "volume": <str>,
    "issn": <str>,
    "pages": <str>,
    "other_ids": {
      "DOI": [
        <str>
      ]
    }
  },
  "BIBREF1": {},
  ...
  "BIBREF25": {}
},
"ref_entries":
  "FIGREF0": {
    "text": <str>, # figure caption text
    "type": "figure"
  },
  ...
  "TABREF13": {
    "text": <str>, # table caption text
    "type": "table"
  }
},
"back_matter": <list of dict> # same structure as body_text
}
}

```

Είναι άξιο να σημειωθεί πως το πεδίο Title, σε μερικά έγγραφα, περιέχει την λέξη “Title” μέσα στο πεδίο του ίδιου του τίτλου, στην αρχή του πεδίου! Για την ώρα η λέξη αυτή παραμένει στην συλλογή.

ΔΗΜΙΟΥΡΓΙΑ ΤΩΝ DOCUMENTS

Αρχικά, σαν κομμάτι του Set-Up αξιοποιούμε ένα JSON Parser για την μετατροπή όλων των JSON αρχείων σε μια πιο απλή TXT μορφή, εύκολη για την ανάγνωση του συστήματος μας για την δημιουργία των Documents. Η αποθήκευση των Documents γίνεται μέσω Scanner στο τοπικό Directory “Documents”, και η συλλέξει των JSON από το τοπικό Directory “Articles”.

>Στην τωρινή μας υλοποίηση, τα πεδία «Τίτλος» και «Κείμενο» συλλέγονται βάση του πεδίου “title” και του πεδίου “text” εντός του “body_text” (βλ. JSON FILES FORMAT). Περαιτέρω πληροφορίες όπως Author μπορούν να προσφέρουν πιο ολοκληρωμένη αναζήτηση στον χρήστη.

Παρακάτω φαίνεται ο τρόπος που η τωρινή μας υλοποίηση παρουσιάζει τον Τίτλο (στην 1^η γραμμή), και το κυρίως κείμενο στην 2^η. (όπως σημειώθηκε πριν, η 2^η γραμμή φτάνει συνήθως πάνω από 60 χιλιάδες χαρακτήρες)

```
1 Live-cell single RNA imaging reveals bursts of translational frameshifting
2 Frameshifting is a fundamental biological process in which a ribosome translating
```

Έχοντας τα αρχείο μας συρρικνωμένα για εύκολη επεξεργασία, περνάμε στο Documentation phase της Lucene. Εκεί δημιουργούμε πολλαπλά Documents με πεδία “title” και “text”, που θα αξιοποιούνται στην αναζήτηση. Η δημιουργία αυτών των Documents μέσω του Standard Analyzer της Lucene, δημιουργεί τα Index με τη χρήση ενός Index Writer και τα τοποθετεί στο τοπικό Directory “Indexes”, και η συλλέξει των Documents γίνεται από τον φάκελο Documents.

Έχουν παρθεί μέτρα για την μπορεί να τρέξει σε διαφορά λειτουργικά συστήματα όσον αναφορά την συλλογή, μετατροπή και αποθήκευση των παραπάνω αρχείων. Η τωρινή υλοποίηση ελέγχει αν έχουν δημιουργηθεί αρχεία ώστε να αποφύγει άσκοπες διεργασίες. Μελλοντικές εκδόσεις πιθανώς να υποστηρίζουν επιλογή αρχείου οπουδήποτε στον δίσκο για όλες τις διεργασίες του Documentation Set-Up, καθώς και την δυνατότητα να αδειάζεις το Database μέσω GUI.

ΑΝΑΖΗΤΗΣΗ

Η διαδικασία εύρεσης γίνεται μέσω ενός Index Searcher. Αξιοποιώντας τα προηγούμενα Index στον φάκελο “Indexes”, καθώς και ένα Query Parser, ελέγχει βάση του Standard Analyzer, τα κείμενα στα Document μας, επιστρέφοντας τα κορυφαία αρχεία βάση ενός score. Μέσω μιας αντίστροφης αναζήτησης, έχοντας τα κλειδιά των κορυφαίων Documents, μπορούμε να βρούμε τα αρχεία και να τυπώσουμε στον χρήστη ότι πληροφορίες χρειάζεται.

>Η τωρινή υλοποίηση αξιοποιεί μόνο το κείμενο για την εύρεση σχετικότερων αρχείων βάση το input του χρήστη. Μελλοντικές εκδόσεις πιθανώς να μπορούν να ελέγχουν τόσο τον τίτλο και τον αρθρογράφο, όσο και να βάζουν αντίστοιχο βάρος στο κάθε πεδίο για πιο στενευμένη αναζήτηση.

Ένα τωρινό αποτέλεσμα της αναζητήσεως στο database μας της λέξης «bees» μας βγάζει τα ακόλουθα άρθρα. Βάση της τωρινής μας υλοποίησης είναι εμφανής ότι τα άρθρα συσχετίζονται με μέλισσες. Πάρα όλα αυτά, αν τοποθετούσαμε παραπάνω βάρος στον τίτλο, το 2^ο αρχείο πιθανώς να ήταν πιο πάνω στην κατάταξη.


```
Welcome to Goggle! What would you like to search?
bees
=====Found the following top 10 results=====
Gut microbiota structure differs between honey
A transmissible RNA pathway in honey bees
The geometry of dependence: solitary bee larvae prioritize carbohydrate over protein in
Gene expression is stronger associated with behaviour than 1 with age and fertility in ant workers 2
Decoupling the effects of nutrition, age and behavioral caste 2 on honey bee physiology and immunity
```

Αναζητώντας την αρχή του κειμένου του προηγούμενου παραδείγματος Document, βλέπουμε πως όντως εμφανίζεται πρώτο όπως περιμέναμε, καθώς και τα υπόλοιπα 9 κορυφαία αρχείο (σε αντίθεση με την πιο σπάνια λέξη “bees”).

```
Welcome to Goggle! What would you like to search?
Frameshifting is a fundamental biological process
=====Found the following top 10 results=====
Live-cell single RNA imaging reveals bursts of translational frameshifting
Extensive programmed ribosomal frameshifting in human as revealed by a massively parallel reporter assay
Cotranslational Folding Stimulates Programmed Ribosomal Frameshifting in the Alphavirus Structural Polyprotein
The translational landscape of Zika virus during infection of mammalian and 1 insect cells 2 3
Comparative analysis of gene expression in virulent and attenuated strains of infectious bronchitis virus at sub-codon resolution
Contesting the evidence for -1 frameshifting in immune-functioning C-C chemokine receptor 5 (CCR5) -the HIV-1 co-receptor 2
Salt Effects on the Thermodynamics of a Frameshifting RNA Pseudoknot under Tension
Crystal structure of Nsp15 endoribonuclease NendoU from SARS-CoV-2 Running Title: Structure of Nsp15 from SARS-CoV-2
A hidden gene in astroviruses encodes a cell-permeabilizing protein involved in virus release
Gene birth contributes to structural disorder encoded by overlapping genes
```

ΑΡΧΙΚΟ GUI ΚΑΙ ΠΑΡΟΥΣΙΑΣΗ ΑΠΟΤΕΛΕΣΜΑΤΩΝ

Η τελική εκδόσει την μηχανής αναζητήσεις θα παρέχει στον χρήστη ένα παράθυρο GUI με πολλαπλά κουμπιά για τις πολλές διεργασίες του συστήματος μας. Μερικά παραδείγματα είναι:

- >Αναζήτηση της λέξης/φράσεις που έχει γραφτεί.
- >Επιλογή φακέλων για την αποθήκευση και φόρτωση της μηχανής (Index/Documents/Articles)
- >Επιλογή αναγραφμένων αρχείων μετά από μια αναζήτηση για περεταίρω διάβασμα των άρθρων.
- >Καθάρισμα των αποτελεσμάτων, η/και των Index/Documents.

Περεταίρω, ένα παράθυρο για το input του χρήστη σε μορφή αλφαριθμητικού για την αναζήτηση. Καθώς και ένα παράθυρο για την διατύπωση των αποτελεσμάτων.

ΔΕΥΤΕΡΗ ΦΑΣΗ ΤΟΥ ΠΡΟΤΖΕΚΤ

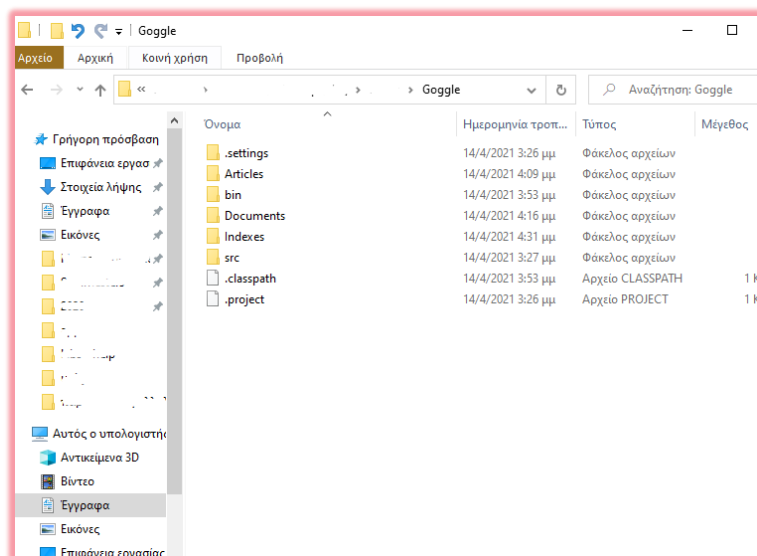
Η ΤΕΛΙΚΗ ΓΡΑΦΙΚΗ ΔΙΕΠΑΦΗ

Στη δεύτερη φάση του πρότζεκτ μας, για να κάνουμε την εφαρμογή μας πιο εύχρηστη για το χρήστη, υλοποιήσαμε ένα απλό παραθυρικό περιβάλλον, στιγμιότυπο του οποίου φαίνεται παρακάτω:



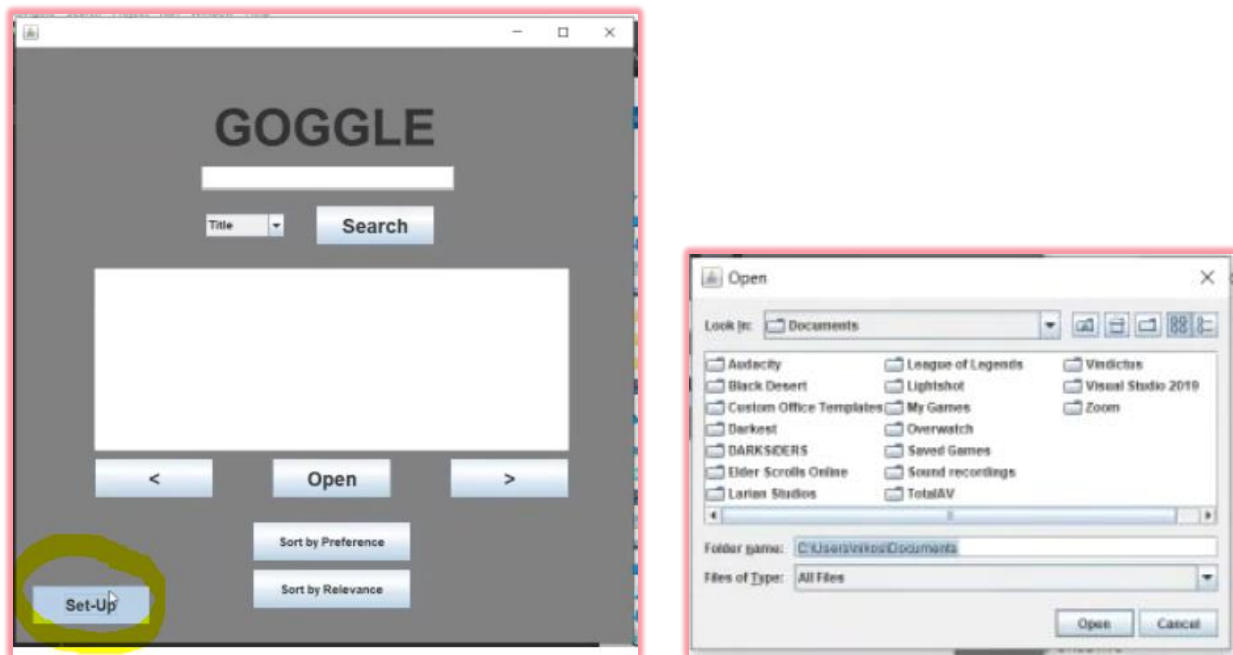
ΤΟ SET-UP ΤΗΣ ΕΦΑΡΜΟΓΗΣ ΜΑΣ

Στον φάκελο “GOGGLE” βρίσκεται ο πηγαίος κώδικας της εφαρμογής μας.



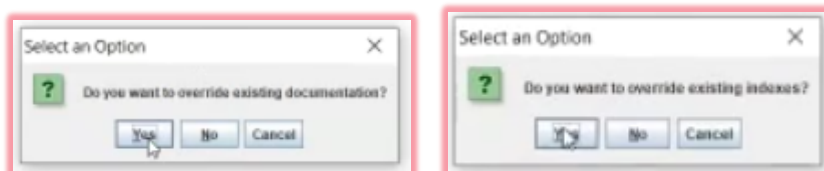
Στην δεύτερη αυτή φάση του πρότζεκτ, και η δημιουργία του CORPUS, έχει πλέον «αυτοματοποιηθεί». Ο χρήστης μόλις

εισέρχεται στην εφαρμογή, χρειάζεται, την πρώτη μόνο φορά, να επιλέξει το πλήκτρο «SET UP» ώστε η Lucene να μπορέσει να κάνει τα μαγικά της!

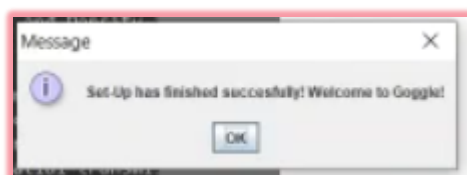


Μόλις επιλέξουμε το κουμπί «Set-Up», πρέπει να επιλέξουμε απλώς τον φάκελο της «πηγής» του Corpus μας, δηλαδή των JSON αρχείων που περιγράψαμε στο πρώτο μέρος του παρόν PDF.

Για να αποφεύγονται ανεπιθύμητες συμπεριφορές από το χρήστη, σε περίπτωση που ήδη έχει πραγματοποιηθεί στον παρόν υπολογιστή το “Set-Up”, το πρόγραμμα εμφανίζει και προειδοποιητικά μηνύματα για να γράψει πάνω στα υπάρχοντα documents και indexes:



Μόλις επιλέξουμε το Set-Up, εάν περιηγηθούμε στο Eclipse, μπορούμε να δούμε πως ξεκινάει η διαδικασία για την δημιουργία του Corpus μας. Αυτή τρέχει στο παρασκήνιο, μέχρι να μας εμφανιστεί μήνυμα επιτυχούς ολοκλήρωσης, ώστε να γνωρίζει ο χρήστης πως η εφαρμογή μας είναι έτοιμη για χρήση:



(Set-Up has finished successfully! Welcome to Goggle!)

Η διαδικασία που τρέχει στο παρασκήνιο περιληπτικά περιγράφεται από δύο βήματα:

Πρώτα από τα JSON αρχεία δημιουργούνται τα απλοποιημένα αρχεία μας:

```
Loading Progress: 18/803
Loading: 18 for source: 07e833d0917cace550853f72923856d0fe1a7120.js
Loading Progress: 19/803
Loading: 19 for source: 08660499ee722a74043f8417faee3e1eeb9d0f5f.js
Loading Progress: 20/803
Loading: 20 for source: 08a22278486e12768ce186677a6a89663a24506f.js
```

Έπειτα αφού ολοκληρωθεί, δημιουργούμε τα indexes:

```
DONE!
Writing 0 : The RNA pseudoknots in foot-and-mouth disease virus an
Writing 1 : Healthcare-resource-adjusted vulnerabilities towards t
Writing 2 : Live-cell single RNA imaging reveals bursts of transla
Writing 3 : Quality control implementation for universal character
```

Να σημειωθεί πως η παραπάνω διαδικασία δημιουργεί δύο καινούργιους φακέλους εντός του φακέλου Goggle:

| | | |
|-----------|--------------------|-----------------|
| Documents | 27/05/2021 3:42 πμ | Φάκελος αρχείων |
| Indexes | 27/05/2021 3:42 πμ | Φάκελος αρχείων |

Επίσης δημιουργεί και ένα αρχείο κειμένου που χρησιμοποιείται για ιστορικό της εφαρμογής μας:

| | | |
|-------------|--------------------|------------------|
| history.txt | 27/05/2021 3:38 πμ | Έγγραφο κειμένου |
|-------------|--------------------|------------------|

Όλα αυτά τα αρχεία δεν χρειάζεται (και δεν πρέπει) να τα πειράζουμε εμείς, υπάρχουν μονάχα για την ορθή λειτουργία της εφαρμογής μας.

ΑΝΑΖΗΤΗΣΗ ΜΕ ΤΟ GOGGLE

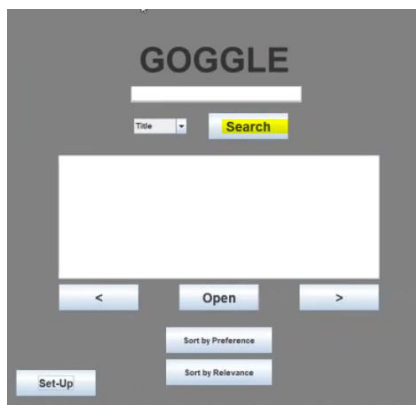
Παρακάτω εξηγούνται τα πεδία της εφαρμογής μας:



← Το πλαίσιο στο οποίο τοποθετούμε τα στοιχεία της αναζήτησής μας (λέξεις κλειδιά)

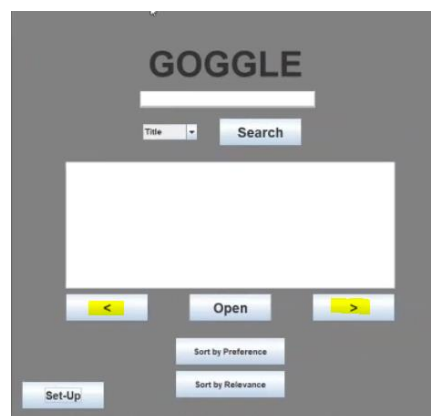
Το drop-down μενού από το → οποίο επιλέγουμε σε ποιο πεδίο θέλουμε να αναζητήσουμε τις λέξεις μας.





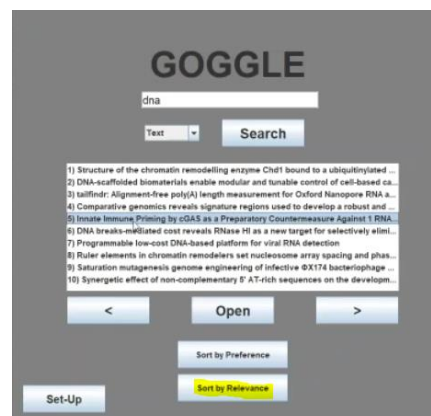
← Το κουμπί έναρξης αναζήτησης που εμφανίζει τα ποιο σχετικά άρθρα. Τα άρθρα εμφανίζονται ανά δέκα σε κάθε σελίδα

Η πλοήγηση μεταξύ των σελίδων → όπου εμφανίζονται τα αποτελέσματα.



← Πλήκτρο ανοίγματος ενός αποτελέσματος σε ξεχωριστό παράθυρο. (open-read)

Πλήκτρο για την εμφάνιση → των αποτελεσμάτων με φθίνουσα σειρά σχετικότητας (relevance) ως προς τους όρους αναζήτησης.




← «SORT BY PREFERENCE», μια επιπλέον λειτουργία την οποία υλοποιήσαμε. Όταν επιλέγεται, τα κείμενα εμφανίζονται με σειρά, η οποία λαμβάνει υπόψιν το ιστορικό αναζήτησής μας και εμφανίζει άρθρα τα οποία, και βέβαια έχουν σχέση με τον όρο που αναζητούμε, αλλά επίσης «προτιμώνται» κείμενα από συγγραφείς των οποίων έχουμε ήδη διαβάσει κάποια κείμενα.

ΑΠΟΚΤΗΣΗ ΤΟΥ GOGGLE & DEMO

Για να λάβει κανείς την εφαρμογή μας αρκεί να περιηγηθεί στο repository μας στο <https://github.com/juulijayrha/lucene>. Κατά την διάρκεια του εαρινού εξαμήνου 2021, δηλαδή μέχρι και τον Ιούνιο, το repository μας θα είναι βέβαια private.

Μέσα στον κατάλόγό μας, παρέχουμε και ένα μικρό βίντεο 4:44 λεπτών το οποίο παρέχει και μια καλύτερη εικόνα λειτουργίας της εφαρμογής μας.

 demo.mp4