

# Εργασία ανάκτησης πληροφορίας

Βασιλειάδης Γιώργος  
ΑΕΜ: 3466

Γραμμένος Θεόδωρος  
ΑΕΜ: 3294

## Χρήση

Το πρόγραμμα είναι διαθέσιμο στο PyPI και μπορεί να εγκατασταθεί με την εντολή `pip install greparl`. Επίσης μπορεί να εκτελεστεί από το φάκελο με τον κώδικα ο οποίος συμπεριλαμβάνεται με την εντολή `python3 -m greparl`. Προϋπόθεση είναι να υπάρχουν στον τρέχοντα φάκελο τα επιπλέον αρχεία με τα μοντέλα κτλ. τα οποία μπορούν να κατέβουν από εδώ. Υποστηρίζονται οι εκδόσεις της Python 3.6 έως 3.9.

## Μέρος I

## Backend

### Γενικές πληροφορίες

Το backend υλοποιείται από τα αρχεία κώδικα που υπάρχουν στο φάκελο `greparl/SearchEngine/backend`. Στο φάκελο `greparl/SearchEngine/preprocessing` βρίσκονται συναρτήσεις που βοηθούν στην επεξεργασία των ομιλιών και στη δημιουργία των απαραίτητων αρχείων για να λειτουργήσει το backend. Καθώς τα αρχεία παίρνουν πολλή ώρα για να δημιουργηθούν υπάρχουν διαθέσιμα στο παραπάνω link. Για τη λειτουργία του προγράμματος αρκεί να γίνει αποσυμπίεση στον root folder που βρίσκεται και ο πηγαίος κώδικας. Εναλλακτικά, τα απαραίτητα αρχεία μπορούν να δημιουργηθούν με τη συνάρτηση `create_all` στο αρχείο `preprocessing/create.py`.

Το αρχείο με τις ομιλίες ονομάζεται `speeches.csv` και βρίσκεται και αυτό στον `root` φάκελο.

## 1 Preprocessing

**funcs.py:** Περιέχει συναρτήσεις που επεξεργάζονται το κείμενο. Ειδικότερα, αφαιρούν τους τόνους, τα σημεία στίξης και κάνουν stem τις λέξεις. Όλα αυτά συνδυάζονται στη συνάρτηση `process_raw_speech_text`, η οποία δέχεται το κείμενο μίας ομιλίας και πραγματοποιεί tokenization, “καθάρισμα” του κειμένου και προαιρετικά stemming και αφαίρεση stopwords. Επίσης υπάρχει η συνάρτηση `process_csv_line` η οποία δέχεται μία γραμμή από το αρχείο csv και εξάγει τα δεδομένα σε ένα αντικείμενο `Speech`.

**extract\_processed\_speeches.py** Περιέχει μία συνάρτηση η οποία δημιουργεί ένα αρχείο το οποίο περιέχει τις επεξεργασμένες ομιλίες, στις οποίες έχει γίνει stemming. Υπάρχει 1-1 αντιστοιχία μεταξύ των ομιλιών στο αρχικό αρχείο και σε αυτό το αρχείο. Δηλαδή, η επεξεργασμένη έκδοση μιας ομιλίας που βρίσκεται στη γραμμή 100 του αρχικού αρχείου, βρίσκεται στη γραμμή 100 του αρχείου που δημιουργείται. Το αρχείο αυτό είναι αναγκαίο καθώς το stemming παίρνει πολλή ώρα, οπότε δημιουργώντας αυτό το αρχείο το κόστος πληρώνεται μόνο μία φορά.

**create.py** Τα αρχεία που το όνομά τους ξεκινάει με `create` περιέχουν τις συναρτήσεις που δημιουργούν τα απαραίτητα αρχεία για τις διάφορες λειτουργίες του προγράμματος.

## 2 Backend

Το backend αποτελείται από ανεξάρτητα modules τα οποία επιτελούν τις ξεχωριστές λειτουργίες του προγράμματος και συνδυάζονται από την κλάση `SpeechBackend`, η οποία αποτελεί το interface του backend.

## 2.1 SpeechFile

Η κλάση `SpeechFile` αναπαριστά ένα αρχείο με ομιλίες. Στο backend κάθε ομιλία αποκτά ένα `id`. Τα `id` ξεκινάνε από το 0 και αυξάνονται με τη σειρά που βρίσκονται οι ομιλίες στο αρχείο `speeches.csv`. Δηλαδή η ομιλία με `id` 0 βρίσκεται στη δεύτερη γραμμή του `speeches.csv` (η πρώτη γραμμή είναι η περιγραφή των πεδίων του csv). Κατά τη δημιουργία του το αντικείμενο κάνει ένα γρήγορο πέρασμα σε όλο το αρχείο και σημειώνει τα `offsets` ανα 100 ομιλίες. Με αυτόν τον τρόπο είναι δυνατή η γρήγορη τυχαία πρόσβαση στις ομιλίες αφού θα χρειαστεί, στη χειρότερη περίπτωση, να διαβαστούν 99 ομιλίες. Δηλαδή, για τις 1.200.000 ομιλίες που υπάρχουν στο αρχείο αποθηκεύονται 12.000 `offsets`. Παρέχονται 2 μέθοδοι για λήψη ομιλιών:

- **`get_speech`** Διαβάζει μία ομιλία από το αρχείο με βάση το `id` της.
- **`get_speeches`** Διαβάζει τις ομιλίες με το δοσμένο `id`. Είναι βελτιστοποιημένη για να φέρνει πολλά έγγραφα, καθώς κάνει επιπλέον υπολογισμούς έτσι ώστε να αποφύγει τα διπλά περάσματα. Για παράδειγμα για τα έγγραφα 508, 509, 550, αν καλούνταν 3 φορές η `get_speech` θα έβρισκε κάθε φορά το `offset` του εγγράφου 500 και θα διάβαζε μέχρι να βρει το έγγραφο με το δοσμένο `id`. Δηλαδή τα έγγραφα 500-507 θα διαβάζονταν 3 φορές. Η `get_speeches` θα έβρισκε το `offset` του εγγράφου 500 και θα έκανε ένα πέρασμα έως το έγγραφο 550, τοποθετώντας τα σχετικά έγγραφα στα αποτελέσματα.

Επίσης παρέχεται μια συνάρτηση `speeches()` που επιστρέφει έναν `iterator` πάνω σε όλες τις ομιλίες, για γρήγορο διάβασμα όλου του αρχείου.

## 2.2 Modules

### 2.2.1 Inverted Index

Ο κώδικας βρίσκεται στο φάκελο `backend/inverted`. Τα αρχεία του `index` αποθηκεύονται στο φάκελο `index`. Το `inverted index` αναπαρίσταται από την κλάση `InvertedIndex`. Το `index` αποτελείται από τρία αρχεία:

1. **`index`** Περιέχει τις λίστες εμφάνισης κάθε όρου. Κάθε γραμμή έχει τη μορφή:

όρος, document\_id\_1, term\_frequency\_1, document\_id\_2,  
term\_frequency\_2 κ.ο.κ

2. **index-catalog** Περιέχει τα offsets της λίστας εμφάνισης κάθε όρου στο αρχείο index. Διατηρείται στη μνήμη.
3. **index-lengths** Περιέχει το μέτρο κάθε εγγράφου, χρησιμοποιείται για τον υπολογισμό του σκορ κάθε εγγράφου. Διατηρείται στη μνήμη.

**Δημιουργία** Η δημιουργία του index γίνεται μέσω της μεθόδου `populate_index`. Η μέθοδος δέχεται έναν πίνακα με τις εμφανίσεις του κάθε όρου σε κάθε document, και ένα array με τα ονόματα των όρων. Και τα 2 μπορούν να δημιουργηθούν μέσω ενός `CountVectorizer` από το `scikit-learn`.

**Αναζήτηση** Η αναζήτηση γίνεται μέσω της μεθόδου `search`, οποία δέχεται κάποια tokens και επιστρέφει τα id των top-k σχετικών εγγράφων. Η αναζήτηση και η βαθμολόγηση γίνεται μέσω του αλγορίθμου αναζήτησης σε `inverted index` που είδαμε στο μάθημα.

**Μέγεθος** Το `inverted index` που δημιουργήθηκε έχει μέγεθος 718 MB. Το μεγαλύτερο μέγεθος καταλαμβάνεται από το αρχείο `index` (676MB) και τα αρχεία `index-catalog` και `index-lengths` καταλαμβάνουν περίπου 40MB.

### 2.2.2 Ομαδοποίηση

Ο κώδικας του συστήματος ομαδοποίησης βρίσκεται στο αρχείο `backend/top/group_manager.py`. Τα αρχεία των ομαδοποιήσεων αποθηκεύονται στο φάκελο `groups`. Κάθε αρχείο στο φάκελο `groups` αντιστοιχεί σε μία ομαδοποίηση των ομιλιών με βάση ένα attribute (π.χ. κόμμα, όνομα ομιλιτή). Το αρχείο για κάθε attribute περιέχει γραμμές της μορφής

attribute\_value, document\_id1, document\_id2 κ.ο.κ.

**Δημιουργία** Η δημιουργία των αρχείων ομάδων γίνεται από τη συνάρτηση `create_groups` στο αρχείο

`preprocessing/create_group.py`. Η συνάρτηση δέχεται ένα αντικείμενο `SpeechFile` και μία λίστα από συναρτήσεις που αντιστοιχίζουν ένα αντικείμενο `Speech` σε ένα `string`, και αντιπροσωπεύουν τα `attributes` ως προς τα οποία γίνεται η ομαδοποίηση. Για παράδειγμα, μία συνάρτηση `party` μπορεί να αντιστοιχίζει μία ομιλία στο κόμμα του ομιλητή. Συνεπώς, θα δημιουργηθεί ένα αρχείο `party` στο φάκελο `groups` που θα περιέχει για κάθε τιμή του `party`, τα `id` των ομιλιών με αυτή την τιμή.

**Χρήση** Η κλάση `GroupManager` διαβάζει όλα τα αρχεία των ομάδων στη μνήμη. Στη συνέχεια παρέχονται οι μέθοδοι:

- **`get_group_attributes`** Επιστρέφει τα `attributes` ως προς τα οποία υπάρχουν ομαδοποιήσεις.
- **`get_attribute`** Δέχεται ένα `attribute`. Επιστρέφει ένα `dict` που αντιστοιχίζει κάθε τιμή αυτού του `attribute` σε μια λίστα με τα `document_id` των εγγράφων που έχουν αυτήν την τιμή στο `attribute`.

**Μέγεθος** Κάθε αρχείο ομάδας έχει μέγεθος περίπου 9MB.

### 2.2.3 Vectorizer και Transformer

Δημιουργούνται 2 αντικείμενα `CountVectorizer` και `TfidfTransformer` τα οποία έχουν εκπαιδευτεί πάνω στο σύνολο των ομιλιών. Τα αρχεία αποθηκεύονται στο φάκελο `tfidf`. Αποθηκεύονται με τη χρήση του module `pickle` της `python`. Το `CountVectorizer` εκπαιδεύεται πάνω στο σύνολο των εγγράφων, ώστε να αποθηκευτεί το `vocabulary` του. Στη συνέχεια το `TfidfTransformer` εκπαιδεύεται πάνω στο σύνολο των εγγράφων, έτσι ώστε να υπολογιστεί το `IDF` κάθε όρου.

**Δημιουργία** Η δημιουργία γίνεται από τη συνάρτηση `create_transformer_vectorizer` στο αρχείο `preprocessing/create.py`. Η συνάρτηση δέχεται ως όρισμα ένα `SpeechFile`.

**Χρήση** Για να χρησιμοποιθούν, τα αντικείμενα μπορούν να φορτωθούν μέσω της `pickle.load` και στη συνέχεια για τον υπολογισμό της συχνότητας εμφανίσεων και των βαθμολογιών `TF-IDF` μπορούν

να χρησιμοποιηθούν οι μέθοδοι transform που παρέχονται από τα αντικείμενα.

**Μέγεθος** Ο Vectorizer έχει μέγεθος 23MB και ο transformer 8MB.

#### 2.2.4 Keyword Manager

Η κλάση KeywordManager είναι υπεύθυνη για την εύρεση Keywords στα έγγραφα. Ο κώδικας βρίσκεται στο αρχείο `top/keyword_manager.py`. Για το keyword extraction δοκιμάστηκαν διάφορες μέθοδοι όπως οι rake, multi\_rake, yake. Καμία όμως δεν έδινε καλά αποτελέσματα, και οι περισσότερες έπαιρναν πάρα πολύ χρόνο (>2 λεπτά). Συνεπώς, η μέθοδος που χρησιμοποιείται είναι η εξής: Τα έγγραφα που για τα οποία θέλουν να εντοπιστούν τα keywords ενώνονται σε ένα μεγάλο έγγραφο, και στη συνέχεια υπολογίζονται οι τιμές για αυτό το ενωμένο έγγραφο, με βάση τα αρχικά IDF που έχουν υπολογιστεί για το σύνολο των εγγράφων. Η μέθοδος αυτή εκτελείται πολύ γρήγορα και παράγει ικανοποιητικά (αλλά όχι τέλεια) αποτελέσματα.

**Δημιουργία** Η KeywordManager χρειάζεται έναν CountVectorizer και έναν TfidfTransformer, εκπαιδευμένους στο σύνολο των εγγράφων για να υπολογίσει τις βαθμολογίες TF-IDF. Μπορούν να χρησιμοποιηθούν αυτοί που έχουν ήδη δημιουργηθεί.

**Χρήση** Παρέχεται η μέθοδος `get_keywords` που δέχεται το κείμενο στο οποίο πρέπει να εντοπιστούν τα keywords και τον αριθμό των keywords που θα εντοπιστούν. Επιπλέον, παρέχεται η δυνατότητα να προστεθούν επιπλέον stopwords τα οποία δε θα ληφθούν υπόψη κατά την ανάκτηση keywords.<sup>1</sup> Με αυτόν τον τρόπο μπορούν να βελτιωθούν τα αποτελέσματα.

**Μέγεθος** Η κλάση δεν απαιτεί κάποια επιπλέον αρχεία.

---

<sup>1</sup>Τα stopwords πρέπει να είναι στην ίδια μορφή με το vocabulary του vectorizer, στα αρχεία που δίνουμε εμείς αυτή είναι ελληνικά, μικρά, χωρίς τόνους. Για σιγουριά μπορεί να γίνει αντιγραφή από προηγούμενο output.

### 2.2.5 Similarity Manager

Η κλάση `SimilarityManager` βρίσκεται στο αρχείο `similarity/similarity_manager.py` και είναι υπεύθυνη για εντοπισμό της ομοιότητας μεταξύ βουλευτών. Χρειάζεται ένα `similarity matrix` το οποίο δείχνει την ομοιότητα μεταξύ των βουλευτών. Το `similarity matrix` αποθηκεύεται στο φάκελο `similarity`. Ο υπολογισμός της ομοιότητας γίνεται με παρόμοιο τρόπο με τον εντοπισμό των `keywords`. Δημιουργείται ένα μεγάλο έγγραφο για κάθε βουλευτή, που περιέχει ενωμένες όλες τις ομιλίες του. Στη συνέχεια υπολογίζονται τα TF-IDF σκορ για κάθε έγγραφο, χρησιμοποιώντας όμως τις IDF τιμές για το σύνολο των ομιλιών. Έπειτα υπολογίζεται η απόσταση συνημιτόνου όλων των ζευγαριών. Οι βουλευτές είναι γύρω στους 1200 οπότε κάτι τέτοιο δεν απαιτεί ιδιαίτερο χρόνο ή χώρο. Επίσης δημιουργείται ένα αρχείο `names.csv` το οποίο περιέχει τα ονόματα των βουλευτών και τα κόμματα στα οποία άνηκαν, για να βοηθήσει στην ανάλυση των αποτελεσμάτων.

Ορίζονται επίσης 2 νέες κλάσεις.

- Η **`SimilarityMember`** περιέχει το όνομα ενός βουλευτή και τα κόμματα των οποίων ήταν μέλος.
- Η **`SimilaritResult`** περιέχει το `original_member` που είναι ένα αντικείμενο `SimilarityMember` για τον βουλευτή για τον οποίο έγινε αναζήτηση, ένα `list similar_members` με τους βουλευτές που είναι παρόμοιοι με τον `original` και μια λίστα `scores` με τα σκορ ομοιότητας του `original` με αυτούς στο `similar_members`. (Κάθε στοιχείο στο `similar_members` αντιστοιχεί σε ένα σκορ στο `scores`, στην ίδια θέση, με την ταξινόμηση να γίνεται από τα περισσότερα στα λιγότερα σχετικά αντικείμενα.)

**Δημιουργία** Ο πίνακας δημιουργείται από τη συνάρτηση `create_similarity_matrix` στο αρχείο `create.py`. Η συνάρτηση δέχεται ένα `dict` με τις ομιλίες ομαδοποιημένες ανα βουλευτή, ένα `CountVectorizer` και ένα `TfidfTransformer`. Στη συνέχεια αποθηκεύονται τα αρχεία με τον πίνακα και το αρχείο `names.csv`.

**Χρήση** Η `SimilarityManager` παρέχει 3 μεθόδους:

- **`get_most_similar`**: Εύρεση των top-k ζευγαριών με τη μεγαλύτερη ομοιότητα. Επιστρέφει λίστα με `tuples` που περιέχουν 2

SimilarityMember για τους 2 βουλευτές και 1 float με το similarity score τους.

- **get\_most\_similar\_to:** Εύρεση των k ομοιότερων βουλευτών με κάποιον βουλευτή. Επιστρέφει SimilarityResult. Στο original\_member είναι το όνομα και τα κόμματα του βουλευτή για τον οποίο έγινε η αναζήτηση. Στο similar\_members επιστρέφονται οι παρόμοιοι βουλευτές και στο scores τα αντίστοιχα σκορ ομοιότητας.
- **get\_similarity\_between\_members:** Εύρεση ομοιότητας μεταξύ 2 βουλευτών. Δέχεται ως όρισμα τα ονόματα των 2 βουλευτών. Επιστρέφει SimilarityResult, όπου στο original\_member είναι ο ένας βουλευτής και στα similar\_members και scores υπάρχει από ένα στοιχείο που αντιστοιχεί στον άλλο βουλευτή και στο similarity score τους, αντίστοιχα.

**Μέγεθος** Το similarity matrix είναι περίπου 18 MB και το names.csv 1 MB.

### 2.2.6 LSA

Η κλάση LSAManager είναι υπεύθυνη για την εκτέλεση ερωτημάτων χρησιμοποιώντας τη μέθοδο LSA. Η εκτέλεση SVD για όλη τη συλλογή εγγράφων ήταν αδύνατη, λόγω των μεγάλων απαιτήσεων σε μνήμη, γι' αυτό δημιουργούνται αρχεία για ένα δείγμα από τις ομιλίες. Ο αριθμός των topics είναι 300. Στα αρχεία που δίνονται περιλαμβάνονται 100000 ομιλίες. Τα αρχεία που απαιτούνται αποθηκεύονται στο φάκελο lsa και είναι τα εξής:

- **matrix.pkl** Περιέχει τον πίνακα που αντιστοιχίζει έγγραφα σε topics.
- **svd.pkl** Περιέχει το αντικείμενο TruncatedSVD που χρησιμοποιείται για την αντιστοίχιση queries σε topics.
- **translation** Αντιστοιχίζει τις σειρές του πίνακα matrix στα id των ομιλιών.
- **vectorizer.pkl** TfidfVectorizer εκπαιδευμένος στα stemmed έγγραφα. Χρησιμοποιείται για την επεξεργασία των queries.



Τα αρχεία για την επεξεργασία ερωτημάτων με LSA απαιτούν πολύ χώρο και συνεπώς δεν διατηρούνται στη μνήμη, αλλά διαβάζονται κάθε φορά που εκτελείται ένα ερώτημα. Επίσης, η επεξεργασία ενός ερωτήματος απαιτεί των υπολογισμό αποστάσεων συνημιτόνου με όλα τα έγγραφα της βάσης. Για αυτούς τους λόγους η επεξεργασία ερωτημάτων με LSA είναι πιο αργή από τον ανεστραμμένο κατάλογο.

**Δημιουργία** Η δημιουργία γίνεται στο αρχείο `create_lsa.py` με τη συνάρτηση `create_sampled_lsa`. Η συνάρτηση δέχεται ως όρισμα μία ομαδοποίηση όλων των ομιλιών, το αρχείο με τις επεξεργασμένες ομιλίες και τον αριθμό των ομιλιών που θα ληφθούν ως δείγμα. Το δείγμα λαμβάνεται με τέτοιο τρόπο ώστε το ποσοστό των ομιλιών να είναι αντιπροσωπευτικό της κάθε ομάδας στο σύνολο των ομιλιών. Δηλαδή, εάν το 50% των ομιλιών ανήκει σε μία ομάδα, το 50% των ομιλιών του δείγματος θα ανήκει σε αυτή την ομάδα/ Η προεπιλεγμένη ομαδοποίηση που γίνεται από την συνάρτηση `create_all` είναι κατά έτος.

**Χρήση** Η μέθοδος `search` είναι αντίστοιχη με αυτή της κλάσης `InvertedIndex`. Δέχεται, δηλαδή, τα `tokens` του ερωτήματος και τον αριθμό των εγγράφων και επιστρέφει τα `top-k` σχετικά έγγραφα, υπολογίζοντας την ομοιότητα συνημιτόνου του ερωτήματος με όλα τα έγγραφα.

**Μέγεθος** Το συνολικό μέγεθος των αρχείων είναι 562 MB, με τα 555 να είναι από τα **`matrix.pkl`** και **`svd.pkl`**.

### 2.2.7 Μοντέλο πρόβλεψης

Για το πέμπτο ερώτημα δημιουργήθηκε ένα μοντέλο το οποίο προσπαθεί να προβλέψει το κόμμα του ομιλητή από μία ομιλία. Ο κώδικας για τη δημιουργία βρίσκεται στο αρχείο `create_ai.py`. Η συνάρτηση `create_sampled_model` δημιουργεί ένα μοντέλο το οποίο προβλέπει μία ιδιότητα ενός `speech` (η προεπιλεγμένη είναι το κόμμα του ομιλητή). Για το μοντέλο έγινε πάλι δειγματοληψία, όπως και στη μέθοδο LSA, καθώς δεν επαρκούσε η RAM του συστήματος για την εκπαίδευσή του στο σύνολο των ομιλιών. Η δειγματοληψία γίνεται με την ίδια μέθοδο που γίνεται στην μέθοδο LSA. Για τον κώδικα, βασιστήκαμε σε αυτόν του εργαστηρίου. Κατά την εκπαίδευση το μοντέλο ήταν 40%

αποτελεσματικό. Έχουμε παρατηρήσει ότι το μοντέλο τείνει να χαρακτηρίζει ότι οι ομιλίες ειπώθηκαν από τη Νέα Δημοκρατία, κάτι που ίσως έχει σχέση με το ότι η Νέα Δημοκρατία έχει το μεγαλύτερο αριθμό ομιλιών στο αρχείο.

**Χρήση** Η μέθοδος `predict_party` στο `SpeechBackend` δέχεται ως όρισμα ένα κείμενο και επιστρέφει το μέλος ποιου κόμματος πιστεύει ότι έκανε αυτήν την ομιλία.

**Μέγεθος** Το μέγεθος του μοντέλου ήταν αρκετά μεγάλο, γι' αυτό εφαρμόζεται συμπίεση η οποία ρίχνει το μέγεθος στα 90 MB.

## 2.3 Χρόνοι δημιουργίας

Ο χρόνος για να δημιουργηθούν όλα τα παραπάνω αρχεία είναι γύρω στις 2.5-3 ώρες. Οι χρόνοι αναφέρονται σε υπολογιστή με επεξεργαστή i5-6500, 16GB μνήμης RAM και δίσκο SSD. Ενδεικτικοί χρόνοι:

- Το `index` δημιουργείται αφού πρώτα έχει δημιουργηθεί το ενδιάμεσο αρχείο με τις επεξεργασμένες ομιλίες στις οποίες, μεταξύ άλλων, έχει πραγματοποιηθεί και `stemming`. Ο `stemmer` που χρησιμοποιείται είναι αρκετά αργός <sup>2</sup> με αποτέλεσμα αυτή η διαδικασία να είναι η πιο αργή κάνοντας γύρω στις 1.5-2 ώρες. Υπάρχει επιτάχυνση από την χρήση πολλαπλών διεργασιών, ταυτόχρονα όμως αυξάνονται πολύ οι απαιτήσεις σε μνήμη. Γι' αυτό το λόγο οι ομιλίες επεξεργάζονται ανα `batches` και γράφονται στο δίσκο, ώστε η χρήση μνήμης να περιοριστεί όσο γίνεται.
- Η δημιουργία των ομάδων γίνεται σε μικρό χρονικό διάστημα, αφού απαιτεί μόνο ένα πέρασμα από το αρχείο των ομιλιών.
- Ο `Vectorizer` και ο `Transformer` δημιουργούνται σε 5-10 λεπτά.
- Το `similarity matrix` δημιουργείται σε 15-20 λεπτά.
- Τα αρχεία για LSA με 100000 έγγραφα και 300 `topics` δημιουργούνται σε 1.5 λεπτό.
- Το μοντέλο με 100000 ομιλίες εκπαιδεύεται σε περίπου μισή ώρα.

---

<sup>2</sup>Σε 1000 ομιλίες η επεξεργασία χωρίς `stemmer` πήρε 1.1s και με `stemmer` πήρε 16.2s. Με τη χρήση `multiprocessing` αυτό μειώθηκε σε 5.2s

## Μέρος II

# Frontend

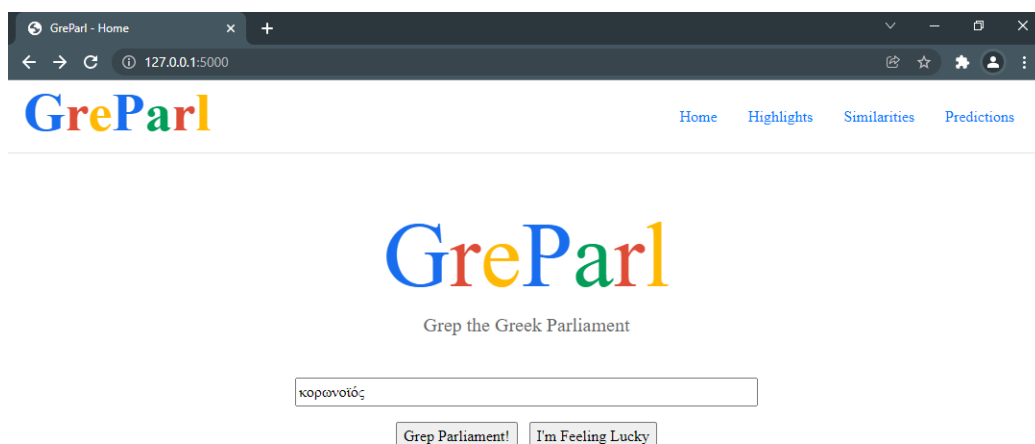
Για την ανάπτυξη της γραφικής διεπαφής της μηχανής αναζήτησης αξιοποιήθηκαν τεχνολογίες δικτύου (HTTP, REST API, HTML/CSS/JS), η βιβλιοθήκη στοιχείων Bootstrap καθώς και το πλαίσιο Flask. Η εφαρμογή αναπτύχθηκε για να εκτελείται τοπικά μέσω WCGI αλλά με μικρή παραμετροποίηση μπορεί να υποστηρίξει και το μοντέλο απομακρυσμένου Client-Server.

Στη συνέχεια επεξηγείται ο τρόπος χρήσης της γραφικής διεπαφής που παρέχεται με την μηχανή αναζήτησης. Για κάθε βασική λειτουργία παρέχεται η αντίστοιχη ενδεικτική μεθοδολογία και ένα εποπτικό σενάριο χρήσης.

### 3 Αναζήτηση

Η αρχική και βασική ενέργεια του χρήστη είναι η αναζήτηση πληροφορίας στο Ελληνικό Κοινοβούλιο. Η αρχική σελίδα προτρέπει τον χρήστη να κάνει ακριβώς αυτό. Ο χρήστης μπορεί να εισάγει το επιθυμητό query και να επιλέξει "Grep Parliament!" για να πραγματοποιήσει την αναζήτηση.

Εναλλακτικά, ο χρήστης μπορεί να εξετάσει ένα τυχαίο δείγμα της διαθέσιμης πληροφορίας επιλέγοντας το γνώριμο "I'm Feeling Lucky".



### 3.1 Περιήγηση στα Αποτελέσματα

Η μηχανή αναζήτησης δύναται να επιστρέψει ένα πλήθος σελίδων πληροφορίας που ταιριάζουν στις απαιτήσεις του χρήστη. Ο χρήστης μπορεί να επιλέξει ποια σελίδα θα επιλέξει εξετάζοντας τις μεταπληροφορίες της εκάστοτε σελίδας. Οι διαθέσιμες μεταπληροφορίες αφορούν την ημερομηνία δημοσίευσης της αντίστοιχης ομιλίας, τον αυτόματα παραγόμενο τίτλο της ομιλίας, και ένα μικρό απόσπασμα της ομιλίας.

Ας σημειωθεί ότι, στην τρέχουσα έκδοση της μηχανής αναζήτησης, δεν υποστηρίζεται η σελιδοποίηση. Συνεπώς, ο χρήστης λαμβάνει πάντα τα καλύτερα 10 (το πολύ) αποτελέσματα στην αναζήτησή του. Αυτός ο περιορισμός αίρεται εύκολα με απευθείας προγραμματιστική χρήση της μηχανής αναζήτησης (backend) αλλά αναμένεται να υποστηριχθεί και γραφικά σε επόμενες εκδόσεις.

A screenshot of a web browser showing the GreParl search results page. The browser's address bar displays '127.0.0.1:5000/search'. The GreParl logo is at the top left, and navigation links for Home, Highlights, Similarities, and Predictions are at the top right. The main heading is 'Results for "κορωνοϊός"', followed by 'About 10 results (0.06 seconds)' and a link to 'Perform a deeper search'. Three search results are listed, each with a blue title and a date: 'Ένεκα Κορωνοϊού' (2020-05-29), 'Όχι «Μπε» Σε Μένα Εσύ Που Πούλαγες Αλοιφές Για Τον Κορωνοϊό!' (2020-05-05), and another identical result.

### 3.2 Βαθιά Αναζήτηση LSA

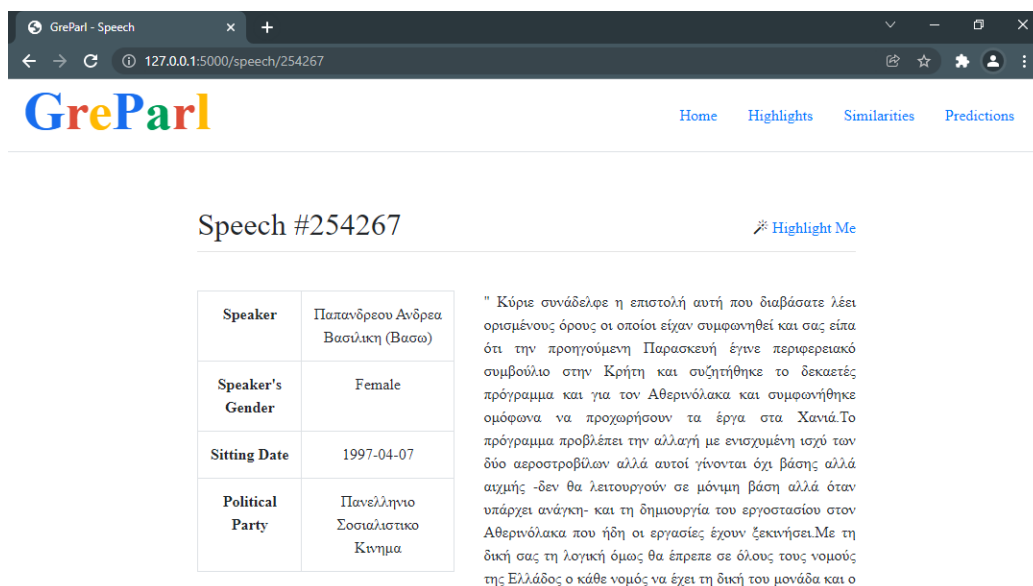
Είναι αναμενόμενο ότι τα αποτελέσματα της αναζήτησης δεν θα ικανοποιούν πάντα τις απαιτήσεις του χρήστη. Η βαθιά αναζήτηση επιτρέπει στον χρήστη να καταβάλει μία δεύτερη, βαθύτερη αναζήτηση χρησιμοποιώντας την επιλογή "Perform a deeper search".

Η βαθύτερη αναζήτηση βασίζεται στην τεχνική LSA και επομένως επιστρέφει αποτελέσματα σχετικά και όχι κατ' ανάγκη ίδια με το query.

A screenshot of the GreParl search results page after clicking 'Perform a deeper search'. The address bar now shows '127.0.0.1:5000/deep-search'. The page structure is identical to the first screenshot, but the search results are different. The three results listed are: 'Η Ιατρική Αμέλεια...' (2017-08-02), 'Περιφερειακό Νοσοκομείο' (2001-02-28), and 'Απολύσατε 1' (1990-12-20).

## 4 Επισκόπηση Ομιλίας

Η πραγματική πληροφορία έγκειται στα περιεχόμενα της ομιλίας. Ο χρήστης μπορεί να διαβάσει μία ομιλία καθώς και τις βασικότερες μεταπληροφορίες της επιλέγοντας κάποιο από τα αποτελέσματα μίας αναζήτησης ή πραγματοποιώντας μία τυχαία αναζήτηση.



The screenshot shows a web browser window with the address bar displaying "127.0.0.1:5000/speech/254267". The page title is "GreParl - Speech". The GreParl logo is visible in the top left, and navigation links for Home, Highlights, Similarities, and Predictions are in the top right. The main content area displays "Speech #254267" with a "Highlight Me" button. Below this is a table with speaker information and a text block with a quote.

Speaker	Παπανδρεου Ανδρεα Βασύλκη (Βασω)
Speaker's Gender	Female
Sitting Date	1997-04-07
Political Party	Πανελληνιο Σοσιαλιστικο Κίνημα

" Κύριε συνάδελφε η επιστολή αυτή που διαβάσατε λέει ορισμένους όρους οι οποίοι είχαν συμφωνηθεί και σας είπα ότι την προηγούμενη Παρασκευή έγινε περιφερειακό συμβούλιο στην Κρήτη και συζητήθηκε το δεκαετές πρόγραμμα και για τον Αθρινόλακα και συμφωνήθηκε ομόφωνα να προχωρήσουν τα έργα στα Χανιά. Το πρόγραμμα προβλέπει την αλλαγή με ενισχυμένη ισχύ των δύο αεροστροβύλων αλλά αυτοί γίνονται όχι βάσης αλλά αιχμής -δεν θα λειτουργούν σε μόνιμη βάση αλλά όταν υπάρχει ανάγκη- και τη δημιουργία του εργοστασίου στον Αθρινόλακα που ήδη οι εργασίες έχουν ξεκινήσει. Με τη δική σας τη λογική όμως θα έπρεπε σε όλους τους νομούς της Ελλάδος ο κάθε νομός να έχει τη δική του μονάδα και ο

## 5 Keywords

Ο χρήστης μπορεί να υπογραμμίσει μία ή περισσότερες ομιλίες εντοπίζοντας τις σημαντικότερες λέξεις που περιέχονται.

Στην απλούστερη περίπτωση υπογράμμισης μίας ομιλίας, ο χρήστης επιλέγει "Highlight Me" στην επισκόπηση μίας ομιλίας. Κατόπιν οδηγείται στην επιλογή παραμέτρων υπογράμμισης, απ' όπου μπορεί να καθορίσει το πλήθος των υπογραμμίσεων.

The screenshot shows a web browser window with the GreParl Highlights page. The URL is 127.0.0.1:5000/highlights/values. The page has a navigation bar with links: Home, Highlights, Similarities, and Predictions. The main heading is "Highlight the Greek Parliament". Below it, the text "Specify options for Speech" is displayed. A form contains two input fields: "#Speech" with the value "254267" and "Results Count" with the value "25". A "Highlight" button is located at the bottom right of the form.

Εναλλακτικά, ο χρήστης μπορεί να καθορίσει ένα σύνολο ομιλιών για τις οποίες θα βρει τις λέξεις κλειδιά. Αυτό μπορεί να γίνει επιλέγοντας "Highlights" από το μενού πλοήγησης. Στη συνέχεια, καλείται να επιλέξει την παράμετρο ομαδοποίησης όπου, επίσης, μπορεί να επιλέξει τις ομιλίες ενός βουλευτή ή ενός πολιτικού κόμματος. Ας σημειωθεί ότι μπορεί να επιλέξει και την τετριμμένη ομαδοποίηση κατά ομιλία.

The screenshot shows a web browser window with the GreParl Highlights page. The URL is 127.0.0.1:5000/highlights/attribute. The page has a navigation bar with links: Home, Highlights, Similarities, and Predictions. The main heading is "Highlight the Greek Parliament". Below it, the text "Take a look at the most common words that are used by Parliament members" is displayed. A form contains a "Search by" label and a dropdown menu with "Party" selected. A "Choose" button is located at the bottom right of the form.

Σε περίπτωση που ο χρήστης επιλέξει να υπογραμμίσει ένα πλήθος

ομιλιών, μπορεί να περιορίσει το χρονικό διάστημα της διαδικασίας, επιταχύνοντας την υπογράμμιση.

The screenshot shows a web browser window with the address bar displaying '127.0.0.1:5000/highlights/values'. The page title is 'GreParl - Highlights'. The GreParl logo is visible in the top left, and navigation links for 'Home', 'Highlights', 'Similarities', and 'Predictions' are in the top right. The main heading is 'Highlight the Greek Parliament', followed by the subtitle 'Specify options for Party'. Below this is a form with four fields: 'Party' (a dropdown menu showing 'Το Ποταμι'), 'From' (a date picker showing '01-Jan-2015'), 'To' (a date picker showing '01-Jan-2020'), and 'Results Count' (a text input showing '25'). A 'Highlight' button is located at the bottom right of the form.

Τελικά, τα αποτελέσματα της υπογράμμισης παρουσιάζονται σε ένα συνοπτικό πίνακα. Οι κυριότερες λέξεις της υπογράμμισης ταξινομούνται από τη σημαντικότερη στη λιγότερο σημαντική.



## Highlight the Greek Parliament

Top 25 Highlights of "Το Ποτάμι" from 2015-01-01  
to 2020-01-01

#	Highlight
1	Κυβέρνηση
2	Ευρώ
3	Συριζα
4	Χώρα
5	Υπουργο
6	Θεμα
7	Νομοσχεδιο
8	Ερώτηση
9	Αριθμο
10	Ποταμι
11	Παρων
12	Τροπολογία
13	Γινεται
14	Πλειοψηφια
15	Χρονια
16	Κανει
17	Ευχαριστουμε
18	Επικαιρη
19	Σωμα
20	Εγινε
21	2015
22	2017
23	Βουλευτη
24	Λεπτα
25	Υπουργος

## 6 Ομοιότητες

Μία άλλη περίπτωση χρήσης είναι η ταύτιση και σύγκριση των βουλευτών βάσει των ομιλιών τους. Ο χρήστης μπορεί να συγκρίνει όλους τους βουλευτές μεταξύ τους και να ανακαλύψει ποιοι μοιάζουν περισσότερο. Επίσης, μπορεί να εξετάσει ποιοι βουλευτές μοιάζουν περισσότερο σε έναν συγκεκριμένο βουλευτή. Ακόμα, μπορεί να συγκρίνει απευθείας δύο βουλευτές και να λάβει έναν δείκτη ομοιότητας.

Αρχικά, χρειάζεται να προσδιορίσει τα συγκρινόμενα μέλη, επιλέγοντας "Similarities" στο μενού πλοήγησης.

The screenshot shows a web browser window with the address bar displaying "127.0.0.1:5000/similarities". The page title is "GreParl - Similarities". The GreParl logo is on the left, and navigation links "Home", "Highlights", "Similarities", and "Predictions" are on the right. The main heading is "Similarize the Greek Parliament", followed by the subtitle "Discover which Parliament members share the same opinions, phrases and interests". The form contains two dropdown menus: "Compare" with the selected value "Παπανδρεου Ανδρεα Γεωργι" and "To" with the selected value "Παπανδρεου Γεωργιου Ανδρ". Below these is a "Results Count" field showing the number "1". A "Similarize" button is located at the bottom right of the form.

Κατόπιν, μπορεί να ανασκοπήσει τα αποτελέσματα καθώς και τους δείκτες ομοιότητας.

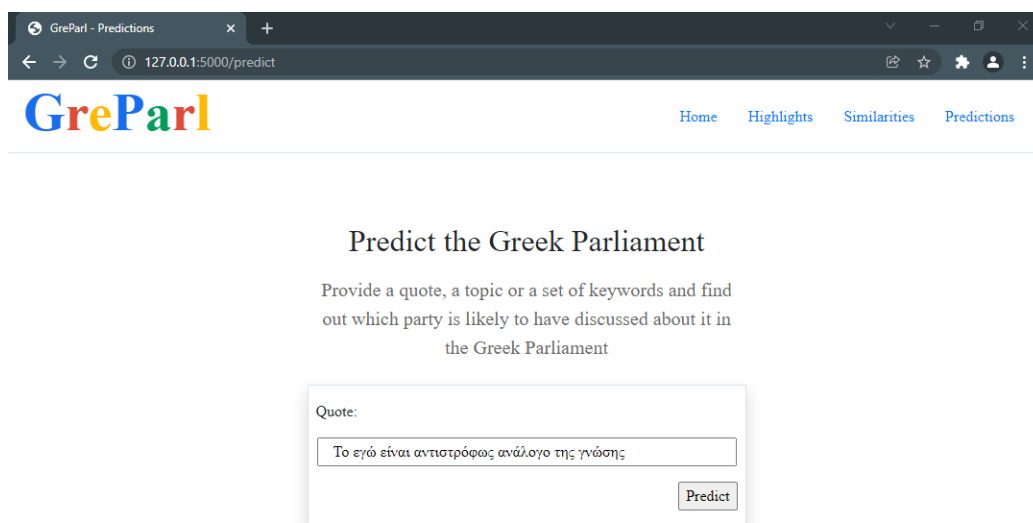
Similarize the Greek Parliament

Top 26 Similarities of Βασιλειαδης Νικολαου Βασίλειος (Λακης)

#	Parliament Member	Similarity
1	Αυγενακης Κωνσταντίνου Ελευθεριος	0.583
2	Τζακρη Εμμανουήλ Θεοδωρα	0.558
3	Μπουγας Δημητριου Ιωαννης	0.549
4	Σταμενιτης Ευαγγελου Διονυσιος	0.548
5	Παπουτσης Πασχαλή Δημητριος	0.543
6	Σκονδρα Κωνσταντίνου Ασημινα	0.541
7	Δερμεντζοπουλος Χρηστου Αλεξανδρος	0.541
8	Κορκα-Κωνστα Αντανιου Αθηνα	0.540
9	Μιχου Κωνσταντίνου Μαρια	0.536
10	Αρβαντιδης Πετρου Γεωργιος	0.532
11	Κεφαλογιαννη Ιωαννη Ολγα	0.529
12	Σκραφνακη Γρηγοριου Μαρια	0.524
13	Λεονταριδης Χρηστου Θεοφύλος	0.523
14	Κοντογιαννης Δημητριου Γεωργιος	0.522
15	Αμμανατιδου-(Πασχαλιδου) Γεωργιου Ευαγγελια (Λιτσα)	0.518
16	Θεοχαρη Αριστοτέλη Μαρια	0.517
17	Γκοκας Ναπολεοντα Χρηστος	0.514
18	Καλαφατης Αθανασιου Σταυρος	0.514
19	Αμμανατιδης Ισαακ Γεωργιος	0.512
20	Στυλιος Δημοσθενη Γεωργιος	0.509
21	Καντερης Ευαγγελου Νικολας	0.509
22	Κατσικης Θεοδωρου Κωνσταντινος	0.508
23	Καραογλου Γεωργιου Θεοδωρος	0.508
24	Βλατης Νικολαου Ιωαννης	0.506
25	Χαρακοπουλος Παντέλη Μαξιμος	0.504
26	Κελλας Αχίλλεα Χρηστος	0.503

## 7 Προβλέψεις

Μεταξύ άλλων, δίνεται η δυνατότητα πρόβλεψης πολιτικών κομμάτων, όπου το σύστημα εκλέγει κάποιο κόμμα βάσει της συνάφειας των ομιλιών του με κάποια δεδομένη φράση. Συγκεκριμένα, ο χρήστης μπορεί να επιλέξει "Predictions" στο μενού πλοήγησης. Στην ανερχόμενη φόρμα, μπορεί να συμπληρώσει ένα σύνολο από βασικές λέξεις ή μία φράση για να αποτελέσει την βάση της εκτίμησης.



GreParl - Predictions

127.0.0.1:5000/predict

Home Highlights Similarities Predictions

### Predict the Greek Parliament

Provide a quote, a topic or a set of keywords and find out which party is likely to have discussed about it in the Greek Parliament

Quote:

Το ερώ είναι αντιστρόφως ανάλογο της γνώσης

Predict

Στη συνέχεια, η μηχανή αναζήτησης θα επιστρέψει τη λίστα των πολιτικών κομμάτων που θεωρεί ότι σχετίζονται περισσότερο με τις δεδομένες λέξεις ή φράσεις.

## Predict the Greek Parliament

Top 1 Predictions of

*Το ερώ είναι αντιστρόφως ανάλογο της  
γνώσης*

#	Predicted Party
1	Πανελλήνιο Σοσιαλιστικό Κίνημα