



ΑΝΑΚΤΗΣΗ ΠΛΗΡΟΦΟΡΙΑΣ

ΠΑΝΕΠΙΣΤΗΜΙΟ ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ/ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ
ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

Ονοματεπώνυμο: Κυριάκος Καμπουρόπουλος

ΑΜ: 20390081

Ονοματεπώνυμο: Δημήτρης Παπούλιας

ΑΜ: 20390183

<https://github.com/Kappakeep/2025>

Βήμα 1: Συλλογή δεδομένων

Τι είναι ένας Web Crawler;

Ένας **Web Crawler** (ή Spider ή Bot) είναι ένα πρόγραμμα ή ένα σενάριο που χρησιμοποιείται για να εξερευνά το διαδίκτυο και να συλλέγει δεδομένα από διάφορες ιστοσελίδες. Ο σκοπός ενός web crawler είναι να "ξεφυλλίσει" τον ιστό, να εντοπίσει και να αποθηκεύσει πληροφορίες που βρίσκονται σε αυτές τις ιστοσελίδες.

Ο τρόπος λειτουργίας του είναι:

1. **Αρχική Ιστοσελίδα (Seed URL):** Ο web crawler ξεκινά από μια αρχική διεύθυνση URL που παρέχεται στον crawler (γνωστή και ως seed URL).
2. **Αναγνώριση Συνδέσμων:** Ο crawler "διαβάζει" το περιεχόμενο της ιστοσελίδας (HTML, συνήθως), εντοπίζει τους συνδέσμους (URLs) που περιέχει η σελίδα και προσθέτει αυτούς τους συνδέσμους στη λίστα των ιστοσελίδων που πρέπει να εξετάσει.
3. **Επισκέπτεται τους Συνδέσμους:** Ο crawler επισκέπτεται αυτούς τους νέους συνδέσμους και επαναλαμβάνει την ίδια διαδικασία: εξαγωγή συνδέσμων, επισκεψη στον ιστότοπο, και ούτω καθεξής.
4. **Αποθήκευση Δεδομένων:** Καθώς ο crawler επισκέπτεται τις ιστοσελίδες, μπορεί να αποθηκεύει δεδομένα όπως τίτλους, κείμενο, εικόνες, συνδέσμους ή άλλες πληροφορίες, για ανάλυση αργότερα. Η αποθήκευση μπορεί να γίνει σε διάφορους τύπους αρχείων, όπως JSON, CSV, ή βάσεις δεδομένων.
5. **Ανανεωμένη Εξέταση (Crawling Frequency):** Οι crawlers μπορεί να τρέχουν περιοδικά για να εντοπίζουν αλλαγές στις ιστοσελίδες ή να εντοπίζουν νέες ιστοσελίδες που εμφανίζονται στο διαδίκτυο.

Στην εργασία μας ο Crawler που υλοποιήσαμε ανακτά 100 άρθρα από το βασικό url της Wikipedia όπως φαίνεται παρακάτω:

```
class WikipediaCrawler:
    def __init__(self, base_url="https://en.wikipedia.org", max_articles=100, keywords=None):
        self.base_url = base_url
        self.visited = set()
        self.max_articles = max_articles
```

Αποφασίσαμε 100 επειδή θεωρήσαμε ότι αυτό είναι ένα καλό δείγμα για να ελέγξουμε την λειτουργικότητα των αλγορίθμων αναζήτησης μας. Δοκιμάστηκαν διαφορετικά σύνολα όπως 1000, αλλά η ώρα ανάκτησης ανερχόταν έως και 30 λεπτά ακόμα και με μικρό περιθώριο timeout. Δώσαμε ένα περιθώριο 2 δευτερολεπτών στο timeout ώστε το πρόγραμμα να προλάβει να διαβάσει τα άρθρα και τα περιεχόμενα τους. Έγιναν δοκιμές με μικρότερα και μεγαλύτερα περιθώρια

όπως 0.2 δευτερόλεπτα το οποίο είχε ως αποτέλεσμα μερικά άρθρα να μην ανακτώνται:

```
else:
    print(f"Failed to fetch {url}, status code: {response.status_code}")
except Exception as e:
    print(f"Error fetching {url}: {e}")
```

Παρακάτω βλέπουμε ένα παράδειγμα εκτέλεσης του κώδικα.

```
Crawled: Artificial intelligence (1/100)
Crawled: Ai (2/100)
Crawled: Artificial intelligence (disambiguation) (3/100)
Crawled: Artificial general intelligence (4/100)
Crawled: Intelligent agent (5/100)
Crawled: Automated planning and scheduling (6/100)
Crawled: Computer vision (7/100)
Crawled: General game playing (8/100)
Crawled: Knowledge representation and reasoning (9/100)
Crawled: Natural language processing (10/100)
Crawled: Robotics (11/100)
Crawled: AI safety (12/100)
Crawled: Machine learning (13/100)
Crawled: Symbolic artificial intelligence (14/100)
Crawled: Deep learning (15/100)
Crawled: Bayesian network (16/100)
Crawled: Evolutionary algorithm (17/100)
Crawled: Hybrid intelligent system (18/100)
Crawled: Artificial intelligence systems integration (19/100)
Crawled: Applications of artificial intelligence (20/100)
Crawled: Machine learning in bioinformatics (21/100)
Crawled: Deepfake (22/100)
Crawled: Machine learning in earth sciences (23/100)
Crawled: Applications of artificial intelligence (24/100)
Crawled: Generative artificial intelligence (25/100)
...
Crawled: Default logic (99/100)
Crawled: Knowledge acquisition (100/100)
Crawling completed. Collected 100 articles.
Data saved to filtered_wikipedia_articles.json
Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...
```

Τι είναι τα Αρχεία JSON;

Τα **JSON (JavaScript Object Notation)** είναι ένα ελαφρύ φορμά για ανταλλαγή δεδομένων, το οποίο είναι εύκολο να διαβαστεί και να γραφτεί από ανθρώπους, αλλά και εύκολο να αναλυθεί και να δημιουργηθεί από υπολογιστές. Είναι ιδιαίτερα

δημοφιλές στον προγραμματισμό ιστού και χρησιμοποιείται για την αποθήκευση και την αποστολή δεδομένων μεταξύ διακομιστών και πελατών, καθώς και για την ανταλλαγή δεδομένων μεταξύ διαφορετικών συστημάτων.


Δομή ενός αρχείου JSON:

- **Αντικείμενα (Objects):** Κλειδιά και τιμές (π.χ., {"name": "John", "age": 30})
- **Πίνακες (Arrays):** Μια λίστα από τιμές (π.χ., ["apple", "banana", "cherry"])

Τα αρχεία JSON χρησιμοποιούνται συνήθως για:

- **Ανταλλαγή δεδομένων μεταξύ διακομιστή και πελάτη.**
- **Αποθήκευση ρυθμίσεων εφαρμογών.**
- **Αποθήκευση δεδομένων σε βάσεις δεδομένων NoSQL (π.χ., MongoDB).**

Παρακάτω βλέπουμε την δημιουργία του .json αρχείου από την εκτέλεση του crawler:

 filtered_wikipedia_articles.json	20/1/2025 4:38 μμ	JSON File	2.880 KB
--	-------------------	-----------	----------

Και το περιεχόμενό του:

```
[
  {
    "title": "Artificial intelligence",
    "url": "https://en.wikipedia.org/wiki/Artificial_intelligence",
    "content": "\n Artificial intelligence (AI), in its broadest sense, is intelligence exhibited by machines, particularly computer systems. It software that enable machines to perceive their environment and use learning and intelligence to take actions that maximize their chances of achieving advanced web search engines (e.g., Google Search); recommendation systems (used by YouTube, Amazon, and Netflix); virtual assistants (e.g., G creative tools (e.g., ChatGPT and AI art); and superhuman play and analysis in strategy games (e.g., chess and Go). However, many AI applications are often without being called AI because once something becomes useful enough and common enough it's not labeled AI anymore.\"[2][3]\n Various subfields The traditional goals of AI research include reasoning, knowledge representation, planning, learning, natural language processing, perception, and su by a human on an at least equal level—is among the field's long-term goals.[4] To reach these goals, AI researchers have adapted and integrated a wide artificial neural networks, and methods based on statistics, operations research, and economics.[b] AI also draws upon psychology, linguistics, philo academic discipline in 1956,[6] and the field went through multiple cycles of optimism throughout its history,[7][8] followed by periods of disappoin increased after 2012 when deep learning outperformed previous AI techniques.[11] This growth accelerated further after 2017 with the transformer arch and the field experienced rapid ongoing progress in what has become known as the AI boom. The emergence of advanced generative AI in the midst of the consequences and harms in the present and raised concerns about the risks of AI and its long-term effects in the future. promoting discussions about
```

Παρατηρούμε ότι έχει σωστή μορφή.

Τι είναι τα Αρχεία CSV;

Τα **CSV (Comma Separated Values)** είναι ένα απλό φoρμά αποθήκευσης δεδομένων σε μορφή πίνακα, όπου κάθε γραμμή αντιπροσωπεύει μία εγγραφή (ή σειρά) και οι τιμές μέσα σε κάθε γραμμή χωρίζονται με κόμμα (ή άλλο χαρακτήρα, όπως το ερωτηματικό ή η tab). Είναι ιδιαίτερα χρήσιμο για την αποθήκευση δομημένων δεδομένων που αντιστοιχούν σε πίνακες (π.χ., δεδομένα βάσεων δεδομένων ή δεδομένα από υπολογιστικά φύλλα).

Δομή ενός αρχείου CSV:

- **Στήλες:** Κάθε στήλη αντιπροσωπεύει μία κατηγορία δεδομένων.
- **Γραμμές:** Κάθε γραμμή αντιπροσωπεύει μία εγγραφή.

Τα αρχεία CSV χρησιμοποιούνται συνήθως για:

- Αποθήκευση και εξαγωγή δεδομένων από βάσεις δεδομένων.
- Ανταλλαγή δεδομένων μεταξύ εφαρμογών που υποστηρίζουν πίνακες.
- Δημιουργία και ανάλυση δεδομένων σε εργαλεία όπως το Excel ή το Google Sheets.

Σύγκριση JSON και CSV:

Χαρακτηριστικό	JSON	CSV
Δομή Δεδομένων	Εξαιρετικά ευέλικτη (μπορεί να περιέχει αντικείμενα και πίνακες)	Απλός πίνακας με σειρές και στήλες
Περίπλοκα Δεδομένα	Υποστηρίζει ιεραρχική δομή (nested data)	Περιορισμένο σε πίνακες
Χρησιμοποιείται για	Ανταλλαγή δεδομένων, Αποθήκευση σε NoSQL βάσεις, API	Εξαγωγή δεδομένων, υπολογιστικά φύλλα
Ευκολία Ανάλυσης	Εύκολο να αναλυθεί με προγραμματισμό	Εύκολο με εργαλεία όπως Excel

Βήμα 2: Προεπεξεργασία κειμένου

Η **προεπεξεργασία κειμένου** (text preprocessing) είναι ένα κρίσιμο στάδιο όταν εργαζόμαστε με δεδομένα φυσικής γλώσσας (Natural Language Processing - NLP). Περιλαμβάνει μια σειρά από τεχνικές που βοηθούν στη βελτίωση της ποιότητας των δεδομένων πριν από την ανάλυση ή εκπαίδευση μοντέλων. Οι εργασίες όπως η **tokenization**, η **stemming/lemmatization**, η **αφαίρεση stop words** και η **αφαίρεση ειδικών χαρακτήρων** είναι απαραίτητες για να εξαγάγουμε χρήσιμα χαρακτηριστικά από το κείμενο και να μειώσουμε την πολυπλοκότητα. Ακολουθεί η αναλυτική εξήγηση του κάθε βήματος:

1. Tokenization (Διαχωρισμός σε Λέξεις/Tokenization)

Τι είναι:

Η tokenization είναι η διαδικασία διαχωρισμού ενός κειμένου σε μικρότερα κομμάτια, που ονομάζονται "tokens". Τα tokens μπορεί να είναι λέξεις, προτάσεις ή ακόμα και χαρακτήρες, ανάλογα με το επίπεδο της tokenization. Στην πράξη, συνήθως χωρίζουμε το κείμενο σε λέξεις ή φράσεις.

Γιατί είναι σημαντική:

- **Αναγνώριση εννοιών:** Κάθε λέξη (ή token) συνήθως μεταφέρει μια σημαντική έννοια, οπότε ο διαχωρισμός σε tokens μας επιτρέπει να επεξεργαστούμε το κείμενο μεμονωμένα για κάθε λέξη ή φράση.

- **Μείωση των δεδομένων:** Κάνει τα δεδομένα πιο εύκολα στη διαχείριση και επιτρέπει την εφαρμογή αλγορίθμων μηχανικής μάθησης ή άλλων τεχνικών ανάλυσης.
- **Εξαγωγή χαρακτηριστικών:** Η tokenization είναι το πρώτο βήμα για την εξαγωγή χαρακτηριστικών από το κείμενο, όπως λέξεις-κλειδιά και συχνότητες λέξεων.

2. Stemming & Lemmatization (Στελέχωση και Λεμματοποίηση)

Τι είναι:

- **Stemming:** Η στελέχωση είναι η διαδικασία αφαίρεσης των καταλήξεων από τις λέξεις για να καταλήξουμε στη βασική τους μορφή, την "ρίζα". Για παράδειγμα, οι λέξεις "running", "runner", "ran" μπορεί να μειωθούν στην ίδια ρίζα "run".
- **Lemmatization:** Η λεμματοποίηση είναι πιο σύνθετη και εξετάζει το λεξικό και τη γραμματική της λέξης, και επιστρέφει τη βασική μορφή της λέξης (λέμα), λαμβάνοντας υπόψη την σημασία και τη σύνταξή της. Για παράδειγμα, η λέξη "better" μπορεί να γίνει "good".

Γιατί είναι σημαντική:

- **Μείωση του μεγέθους του λεξικού:** Αφαιρώντας καταλήξεις και επαναφέροντας τις λέξεις στη ριζική τους μορφή, μειώνουμε τον αριθμό των μοναδικών λέξεων που πρέπει να επεξεργαστούμε.
- **Αποφυγή της υπερβολικής πολυπλοκότητας:** Πολλές λέξεις με παρόμοια σημασία μπορεί να παρατηρηθούν με διαφορετικές γραμματικές μορφές. Η στελέχωση και η λεμματοποίηση βοηθούν να ομαλοποιήσουμε αυτές τις λέξεις, επιτρέποντας στο μοντέλο να αναγνωρίσει ότι ενδέχεται να σημαίνουν το ίδιο πράγμα.
- **Ακρίβεια στην ανάλυση:** Η λεμματοποίηση μπορεί να βοηθήσει στην ακριβέστερη κατανόηση της σημασίας μιας λέξης σε σχέση με το κείμενο, κάτι που είναι χρήσιμο σε πιο προηγμένα μοντέλα ανάλυσης.

3. Stop-word Removal (Αφαίρεση Stop Words)

Τι είναι:

Οι **stop words** είναι λέξεις που εμφανίζονται πολύ συχνά σε ένα κείμενο αλλά δεν προσφέρουν σημαντική πληροφορία για την ανάλυση (όπως "και", "το", "το", "είναι"). Αφαίρεση αυτών των λέξεων μειώνει την ποσότητα δεδομένων χωρίς να χάσουμε χρήσιμη πληροφορία.

Γιατί είναι σημαντική:

- **Μείωση θορύβου:** Οι stop words συνήθως προσθέτουν "θόρυβο" χωρίς να παρέχουν σημαντική έννοια. Η αφαίρεση τους βοηθά στο να επικεντρωθούμε στις σημαντικές λέξεις του κειμένου.
- **Αύξηση της απόδοσης:** Αφαιρώντας τις λέξεις που δεν έχουν αξία, μπορούμε να μειώσουμε την υπολογιστική πολυπλοκότητα των μοντέλων μηχανικής μάθησης και να αυξήσουμε την ακρίβεια.
- **Βελτίωση του προφίλ των χαρακτηριστικών:** Το μοντέλο θα μπορεί να επικεντρωθεί σε λέξεις που συνεισφέρουν περισσότερο στην κατηγοριοποίηση ή ανάλυση.

4. Αφαίρεση Ειδικών Χαρακτήρων (Removing Special Characters)

Τι είναι:

Η αφαίρεση ειδικών χαρακτήρων περιλαμβάνει την αφαίρεση χαρακτήρων όπως σημεία στίξης, αριθμοί, και άλλοι μη αλφαβητικοί χαρακτήρες (π.χ., #, @, \$, %), οι οποίοι μπορεί να μην έχουν σημασία στην ανάλυση του κειμένου.

Γιατί είναι σημαντική:

- **Απλοποίηση των δεδομένων:** Οι ειδικοί χαρακτήρες συχνά δεν προσφέρουν καμία χρήσιμη πληροφορία για την ανάλυση κειμένου (εκτός αν δουλεύουμε με δεδομένα που περιέχουν εκφράσεις ή σύμβολα που είναι σημαντικά).
- **Αποφυγή λάθους ανάλυσης:** Αυτοί οι χαρακτήρες μπορεί να προκαλέσουν σφάλματα ή παρανοήσεις κατά την επεξεργασία του κειμένου, όπως κατά την αναγνώριση λέξεων ή την κατασκευή λεξικών.
- **Καθαρές αναλύσεις:** Η αφαίρεση αυτών των χαρακτήρων επιτρέπει στο σύστημα να εστιάσει στις λέξεις και την πραγματική έννοια του κειμένου, χωρίς να επηρεάζεται από ασήμαντα σύμβολα.

Βήμα 3: Ανεστραμμένο ευρετήριο

Το **ανεστραμμένο ευρετήριο** (ή **inverted index**) είναι μια δομή δεδομένων που χρησιμοποιείται ευρέως σε συστήματα αναζήτησης (search engines) για να βελτιώσει την αποδοτικότητα της αναζήτησης σε μεγάλα σύνολα κειμένων ή εγγράφων. Η βασική ιδέα πίσω από το ανεστραμμένο ευρετήριο είναι να δημιουργηθεί ένας κατάλογος που ανατρέπει τη διαδικασία αναζήτησης: αντί να αναζητούμε τα έγγραφα για συγκεκριμένες λέξεις, αντίθετα καταγράφουμε ποιες λέξεις εμφανίζονται σε ποια έγγραφα.

Ανάλυση του Ανεστραμμένου Ευρετηρίου

Ας υποθέσουμε ότι έχουμε έναν μικρό σύνολο εγγράφων με τις παρακάτω φράσεις:

- **Έγγραφο 1:** "Το σύστημα αναζήτησης είναι γρήγορο."
- **Έγγραφο 2:** "Η αναζήτηση είναι βασικό στοιχείο."

- **Έγγραφο 3:** "Το σύστημα αναζήτησης χρησιμοποιεί αλγόριθμους."

Πλεονεκτήματα του Ανεστραμμένου Ευρετηρίου

- **Γρήγορη Αναζήτηση:** Η βασική πλεονεκτική χρήση του ανεστραμμένου ευρετηρίου είναι ότι επιτρέπει πολύ γρήγορη αναζήτηση, καθώς μπορείς να βρεις άμεσα σε ποια έγγραφα εμφανίζεται μια συγκεκριμένη λέξη, χωρίς να χρειάζεται να σαρώσεις κάθε έγγραφο. Αυτό επιταχύνει κατά πολύ την αναζήτηση.
- **Αποδοτικότητα στην Αποθήκευση:** Αντί να αποθηκεύονται οι λέξεις για κάθε έγγραφο (όπως σε ένα παραδοσιακό ευρετήριο), καταγράφονται οι λέξεις και τα έγγραφα στα οποία εμφανίζονται, γεγονός που μειώνει τη χωρητικότητα της αποθήκευσης.
- **Σύνθετες Αναζητήσεις:** Το ανεστραμμένο ευρετήριο υποστηρίζει την εκτέλεση σύνθετων ερωτημάτων αναζήτησης, όπως οι λογικοί συνδυασμοί (AND, OR, NOT), καθώς μπορούμε να αναζητήσουμε σε ποια έγγραφα υπάρχουν όλες οι λέξεις ή κάποια από αυτές.

Βήμα 4: Μηχανή αναζήτησης (Search Engine)

Ο αλγόριθμος **TF-IDF** (Term Frequency - Inverse Document Frequency) είναι ένας από τους πιο διαδεδομένους αλγόριθμους για την εκτίμηση της "σημαντικότητας" των λέξεων μέσα σε ένα κείμενο ή έγγραφο, σε σχέση με ένα σύνολο εγγράφων. Χρησιμοποιείται συχνά σε εφαρμογές **ανάκτησης πληροφορίας** (information retrieval) και **εξόρυξης κειμένου** (text mining), καθώς βοηθά στην επιλογή των πιο σχετικών λέξεων από ένα σύνολο εγγράφων.

Ο αλγόριθμος TF-IDF βασίζεται σε δύο κυριότερους παράγοντες:

1. **TF (Term Frequency):** Η συχνότητα εμφάνισης μιας λέξης σε ένα έγγραφο.
2. **IDF (Inverse Document Frequency):** Η αντίστροφη συχνότητα εμφάνισης της λέξης σε όλα τα έγγραφα του συνόλου.

Χρήση του TF-IDF

1. **Αναγνώριση Σημαντικών Λέξεων:** Χρησιμοποιώντας το TF-IDF, μπορούμε να αναγνωρίσουμε ποιες λέξεις είναι σημαντικές για ένα έγγραφο. Για παράδειγμα, μια λέξη με υψηλό TF-IDF είναι πιθανότατα πολύ πιο σχετική για το περιεχόμενο του εγγράφου από μια λέξη με χαμηλό TF-IDF.
2. **Συσχέτιση Εγγράφων:** Το TF-IDF χρησιμοποιείται για να υπολογιστεί πόσο σχετικό είναι ένα έγγραφο με μια αναζήτηση. Η αναζήτηση γίνεται με βάση τις λέξεις-κλειδιά και οι λέξεις με το υψηλότερο TF-IDF θεωρούνται πιο σημαντικές και συναφείς με το ερώτημα.

3. **Μηχανισμοί Συνιστώσεων:** Στην εξόρυξη κειμένου και τη μηχανική μάθηση, το TF-IDF μπορεί να χρησιμοποιηθεί για τη δημιουργία συστημάτων συνιστώσεων που συνιστούν έγγραφα ή προϊόντα που είναι πιο σχετικά με τον χρήστη.

```
Results (100 found):
- Neural network (machine learning) (score: 0.4864)
- Deep learning (score: 0.3705)
- Bayesian network (score: 0.1614)
- Generative audio (score: 0.1592)
- Machine learning in bioinformatics (score: 0.1205)
- Artificial intelligence (score: 0.1142)
- History of artificial intelligence (score: 0.1049)
- Machine learning (score: 0.1028)
- Natural language processing (score: 0.1013)
- Symbolic artificial intelligence (score: 0.0974)
- Computational creativity (score: 0.0963)
- Machine learning in earth sciences (score: 0.0759)
- Neuroscience (score: 0.0743)
- Applications of artificial intelligence (score: 0.0686)
- Applications of artificial intelligence (score: 0.0686)
- Applications of artificial intelligence (score: 0.0686)
- Generative artificial intelligence (score: 0.0564)
- AI winter (score: 0.0505)
- Transformer (deep learning architecture) (score: 0.0457)
- Recommender system (score: 0.0422)
- Artificial intelligence art (score: 0.0338)
...
- Knowledge base (score: 0.0000)
- Default logic (score: 0.0000)
- Knowledge acquisition (score: 0.0000)
Exiting the search engine.
```

Παραπάνω είναι τα αποτελέσματα στην αναζήτηση “neural networks” με χρήση του αλγορίθμου TF-IDF. Δεν καταφέραμε να εξαλείψουμε διπλότυπα.

BM25 (Best Matching 25)

Ο αλγόριθμος BM25 είναι μια εξέλιξη του TF-IDF και ανήκει στην κατηγορία των συναρτήσεων κατάταξης. Εφαρμόζεται κυρίως στην ανάκτηση πληροφορίας και χρησιμοποιεί την συχνότητα λέξης (TF) και την αντίστροφη συχνότητα εγγράφου (IDF), αλλά κάνει επίσης κάποιες τροποποιήσεις για να βελτιώσει τα αποτελέσματα της αναζήτησης.

Χαρακτηριστικά BM25:

- **Προσαρμοστικότητα:** Αντιμετωπίζει την αποδυνάμωση της σημασίας των λέξεων όταν αυτές επαναλαμβάνονται πολλές φορές.

- **Αντιμετώπιση μεγάλων εγγράφων:** Η παράμετρος bbb επιτρέπει τη ρύθμιση της "ανάγνωσης" μεγάλων εγγράφων σε σχέση με μικρών εγγράφων.
- **Στατιστικά και Απόδοση:** Χρησιμοποιείται ευρέως σε συστήματα αναζήτησης και έχει επιδείξει εξαιρετική απόδοση σε πολλές μελέτες.

```
Results (100 found):
- Neural network (machine learning) (score: 4.2436)
- Deep learning (score: 4.2204)
- Machine learning in bioinformatics (score: 3.9853)
- Computational creativity (score: 3.9188)
- Machine learning (score: 3.9076)
- Symbolic artificial intelligence (score: 3.8597)
- Artificial intelligence (score: 3.8407)
- Machine learning in earth sciences (score: 3.8316)
- Natural language processing (score: 3.8090)
- History of artificial intelligence (score: 3.8068)
- Transformer (deep learning architecture) (score: 3.7974)
- Neuroscience (score: 3.6999)
- Generative audio (score: 3.6923)
- Recommender system (score: 3.6514)
- Generative artificial intelligence (score: 3.6260)
- Applications of artificial intelligence (score: 3.6118)
- Applications of artificial intelligence (score: 3.6118)
- Applications of artificial intelligence (score: 3.6118)
- AI winter (score: 3.4689)
- Deepfake (score: 3.3426)
- Computer vision (score: 3.3339)
...
- Knowledge base (score: 0.0000)
- Default logic (score: 0.0000)
- Knowledge acquisition (score: 0.0000)
```

Παραπάνω βλέπουμε τα αποτελέσματα στην ερώτηση “neural network” με την χρήση του αλγορίθμου bm25. Παρατηρούμε ότι επιστρέφει αποτελέσματα με αρκετή ακρίβεια. Το ένα πρόβλημα που δεν καταφέραμε είναι να εξαλείψουμε duplicates. (Όπως φαίνεται από το άρθρο Applications of artificial intelligence).

Boolean Search (Αναζήτηση Boolean)

Η **Boolean αναζήτηση** είναι ένας απλός αλγόριθμος ανάκτησης πληροφοριών που χρησιμοποιεί **λογικούς τελεστές** για να καθορίσει πώς να συνδυάσει όρους στην αναζήτηση. Οι κύριοι λογικοί τελεστές που χρησιμοποιούνται είναι οι εξής:

1. **AND:** Βρίσκει έγγραφα που περιέχουν **όλους** τους όρους που αναφέρονται στην αναζήτηση.
2. **OR:** Βρίσκει έγγραφα που περιέχουν **τουλάχιστον έναν** από τους όρους που αναφέρονται.

3. **NOT:** Βρίσκει έγγραφα που **δεν περιέχουν** τον όρο που αναφέρεται.

Χαρακτηριστικά Boolean:

- **Απλότητα και Ευκολία:** Η Boolean αναζήτηση είναι εύκολη στην εφαρμογή και κατανοητή για τους χρήστες.
- **Ακρίβεια:** Ο χρήστης μπορεί να είναι ακριβής στην αναζήτησή του, καθορίζοντας αυστηρά τους όρους και τους λογικούς τελεστές.
- **Περιορισμοί:** Ένα από τα κυριότερα μειονεκτήματα της Boolean αναζήτησης είναι ότι δεν υπολογίζει την **σημαντικότητα** των λέξεων ή τη **σχετικότητα** των εγγράφων, απλά αναζητά τη παρουσία/απουσία τους.

Στην εργασία μας δεν καταφέραμε να υλοποιήσουμε σωστά τον αλγόριθμο Boolean και δεν εμφανιζόντουσαν τα αποτελέσματα που θέλαμε.

Vector Space Model (VSM)

Το Vector Space Model (VSM) είναι ένα μαθηματικό μοντέλο για την αναπαράσταση εγγράφων και όρων σε έναν πολυδιάστατο χώρο. Κάθε έγγραφο αντιπροσωπεύεται ως διάνυσμα λέξεων σε ένα διανυσματικό χώρο, και η κάθε λέξη είναι μια διάσταση. Οι τιμές των διανυσμάτων αυτών συνήθως υπολογίζονται μέσω στατιστικών μέτρων όπως το TF, IDF, ή άλλες εκτιμήσεις.

Χαρακτηριστικά VSM:

- **Διανυσματική Αναπαράσταση:** Οι λέξεις και τα έγγραφα αναπαριστώνται με αριθμητικά διανύσματα.
- **Σχετικότητα:** Μπορεί να υπολογιστεί η σχετικότητα των εγγράφων με το ερώτημα.
- **Περιορισμοί:** Το VSM μπορεί να έχει προβλήματα με τη διαχείριση μεγάλων συνόλων δεδομένων και τη **διάσταση του χώρου** (μεγαλύτερος αριθμός λέξεων = υψηλότερος αριθμός διαστάσεων).

Παρακάτω είναι ένα παράδειγμα αποτελεσμάτων του VSM. Αρκετά ακριβές.

Results (100 found):

- Neural network (machine learning) (score: 0.3248)
- Deep learning (score: 0.2640)
- Generative audio (score: 0.0963)
- Artificial intelligence (score: 0.0932)
- Machine learning (score: 0.0872)
- History of artificial intelligence (score: 0.0787)
- Machine learning in bioinformatics (score: 0.0708)
- Symbolic artificial intelligence (score: 0.0693)
- Natural language processing (score: 0.0682)
- Bayesian network (score: 0.0681)
- Computational creativity (score: 0.0566)
- Applications of artificial intelligence (score: 0.0532)
- Applications of artificial intelligence (score: 0.0532)
- Applications of artificial intelligence (score: 0.0532)
- Neuroscience (score: 0.0501)
- Machine learning in earth sciences (score: 0.0450)
- Generative artificial intelligence (score: 0.0410)
- AI winter (score: 0.0321)
- Recommender system (score: 0.0265)
- Computer vision (score: 0.0246)
- Glossary of artificial intelligence (score: 0.0242)
- ...
- Knowledge base (score: 0.0000)
- Default logic (score: 0.0000)
- Knowledge acquisition (score: 0.0000)

Exiting the search engine.