# Προγραμματιστική Εργασία: Επέκταση Ερωτημάτων με Συνώνυμους Όρους για τη Βελτίωση των Αποτελεσμάτων της Ανάκτησης

Ο σκοπός της εργασίας είναι να εξασκηθείτε σε κλασικές μεθόδους και μοντέλα ανάκτησης πληροφορίας, αλλά και να εφαρμόσετε state-of-the-art τεχνικές για να βελτιώσετε τα αποτελέσματα ενός συστήματος ανάκτησης πάνω σε πραγματικά δεδομένα.

Ένα από τα πιο σημαντικά βήματα δημιουργίας ενός συστήματος ανάκτησης πληροφορίας είναι η εφαρμογή αλγορίθμων ανάλυσης κειμένου στη συλλογή δεδομένων. Οι αλγόριθμοι αυτοί προσδιορίζουν τον τρόπο με τον οποίο γίνεται η επεξεργασία του κειμένου μας και προκύπτουν οι όροι που θα μπουν στο ευρετήριό μας. Το ερώτημα του χρήστη αφού υποβληθεί θα υποστεί αντίστοιχη επεξεργασία και οι όροι του θα συγκριθούν με τους όρους του ευρετηρίου. Τα κείμενα που περιέχουν τους όρους του ερωτήματος θα επιστραφούν ως συναφή στο χρήστη.

Ένα από τα μεγαλύτερα προβλήματα στη διαδικασία αυτή είναι ότι οι χρήστες μπορεί να εκφράσουν την πληροφοριακή τους ανάγκη με διαφορετικούς τρόπους. Για παράδειγμα, το "walk in the mountains" μπορεί να εκφραστεί και ως "trekking" ή "hiking". Αν ο χρήστης υποβάλει το ερώτημα "hiking", αλλά το ευρετήριο περιέχει κείμενα με τη λέξη trekking, τα κείμενα αυτά δεν θα επιστραφούν στο χρήστη, παρόλο που είναι συναφή. Ο χρήστης είναι πιθανό να μην ικανοποιήσει την πληροφοριακή του ανάγκη. Ένας τρόπος για να αντιμετωπιστεί το πρόβλημα αυτό είναι το σύστημα ανάκτησης να γνωρίζει τα συνώνυμα των όρων του ερωτήματος.

Καλείστε να δημιουργήσετε μια μηχανή αναζήτησης, η οποία θα χρησιμοποιεί την τεχνική της επέκτασης-διεύρυνσης του ερωτήματος του χρήστη με συνώνυμους όρους έτσι ώστε να μπορεί να εκφράσει την πληροφοριακή ανάγκη του χρήστη με διαφορετικούς τρόπους και να αντιμετωπίσει προβλήματα όπως αυτό που περιγράφηκε παραπάνω.

Θα εφαρμόσετε δύο τρόπους για να βρίσκετε συνώνυμα. Ο ένας είναι με χρήση ειδικού λεξικού συνώνυμων όρων και ο άλλος με χρήση του δημοφιλή αλγόριθμου νευρωνικών δικτύων word2vec. Κάθε μεθοδολογία έχει τα δικά της πλεονεκτήματα, αλλά και τους δικούς της περιορισμούς.

Τέλος, θα αξιολογήσετε τη μηχανή αναζήτησής σας πάνω στη συλλογή `IR2025` χρησιμοποιώντας το εργαλείο αξιολόγησης `trec_eval`.

**Φάση 1 – Κλασική ανάκτηση (15 μονάδες)**

Στην πρώτη φάση της εργασίας θα εφαρμόσετε ένα από τα κλασικά μοντέλα ανάκτησης που θα δούμε στο μάθημα (πχ. μοντέλο διανυσματικού χώρου, πιθανοτικό μοντέλο, κλπ.) στη συλλογή κειμένων `IR2025`. Η `IR2025` είναι μία συλλογή κειμένων. Περιλαμβάνει επίσης ένα σύνολο από ερωτήματα μαζί με τις σωστές συναφείς απαντήσεις. Το σύστημά σας απλά θα επιστρέφει τα πιο σχετικά κείμενα σε κάθε ερώτημα με βάση το επιλεγμένο μοντέλο ανάκτησης, χωρίς να χρησιμοποιεί την τεχνική της επέκτασης-διεύρυνσης του ερωτήματος του χρήστη με συνώνυμους όρους.

1. Προεπεξεργαστείτε τη συλλογή κειμένων `IR2025` προκειμένου να είναι σε κατάλληλη μορφή για να χρησιμοποιηθεί από τη μηχανή αναζήτησης `ElasticSearch`.
2. Δημιουργήστε ένα ευρετήριο από τη συλλογή χρησιμοποιώντας τη μηχανή αναζήτησης `ElasticSearch`. Επιλέξτε κατάλληλο `Analyzer` και συνάρτηση ομοιότητας που επιθυμείτε. Κάθε κείμενο θα πρέπει να αποθηκευτεί σε ένα `field` της `ElasticSearch`.
3. Εκτελέστε τα ερωτήματα πάνω στο ευρετήριο και συλλέξτε τις απαντήσεις της μηχανής, τα `k` πρώτα ανακτηθέντα κείμενα, για `k = 20, 30, 50`.
4. Αξιολογήστε τις απαντήσεις σας συγκρίνοντάς τες με τις σωστές απαντήσεις χρησιμοποιώντας το εργαλείο αξιολόγησης `trec_eval` και τα μέτρα αξιολόγησης `MAP` (mean average precision) και `avgPre@k` (μέση ακρίβεια στα `k` πρώτα ανακτηθέντα κείμενα) για `k = 5, 10, 15, 20`.
5. Καταγράψτε τα πειράματά σας σε μια αναφορά. Περιγράψτε πώς υλοποιήσατε τα 4 παραπάνω βήματα, συμπεριλάβετε screenshots όπου θεωρείτε χρήσιμο (εγκατάσταση εργαλείων, εκτέλεση κώδικα, εκτέλεση `trec_eval`), και φτιάξτε έναν πίνακα με τα αποτελέσματα του `trec_eval` για τις διάφορες τιμές του `k`. Συζητήστε τα αποτελέσματα σας. Δημιουργήστε ένα αρχείο pdf με την αναφορά σας. Θα υποβάλετε την αναφορά σας, τον κώδικά σας και τα αποτελέσματα του `trec_eval` σε ένα αρχείο `zip` με ονομασία `αριθμός_μητρώου1_αριθμός_μητρώου2.zip` (πχ. 3220100_3220200.zip). Μη συμπεριλάβετε τη συλλογή κειμένων. Καταγράψτε τις πηγές σας.

**Φάση 2 – Επέκταση ερωτήματος με συνώνυμα από το `WordNet` (10 μονάδες)**

Στη δεύτερη φάση της εργασίας θα επεκτείνετε τα ερωτήματα της συλλογής `IR2025` με συνώνυμους όρους που θα αντλήσετε από το `WordNet`. Το `WordNet` είναι μια λεξική βάση δεδομένων για την αγγλική γλώσσα. Ομαδοποιεί τις αγγλικές λέξεις σε σύνολα συνωνύμων που ονομάζονται *synsets*, παρέχει σύντομους ορισμούς και παραδείγματα χρήσης των λέξεων και καταγράφει ορισμένες σχέσεις μεταξύ αυτών των συνόλων ή των μελών τους.

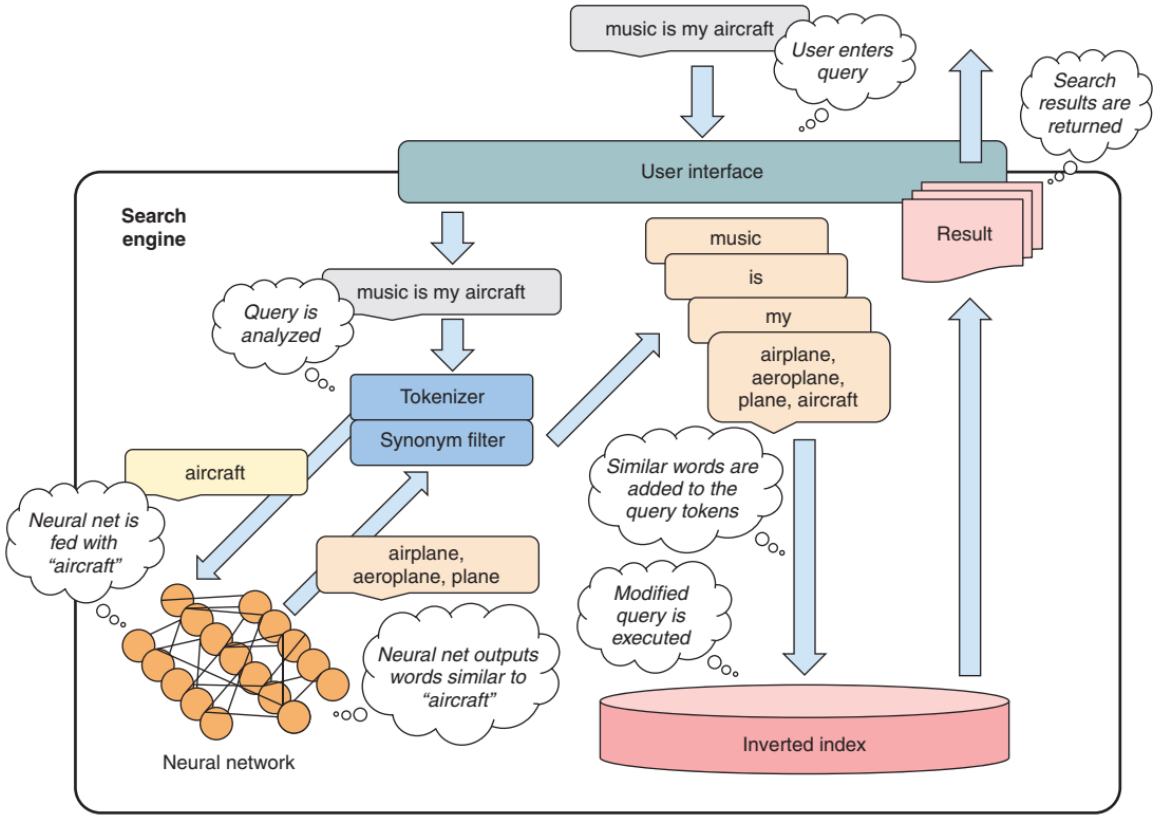
1. Χρησιμοποιήστε τη βιβλιοθήκη `NLTK` για να αντλήσετε συνώνυμα από το `WordNet` (δείτε κώδικα 1.0 για παράδειγμα).
2. Επεκτείνετε τα ερωτήματα της `IR2025` με τους συνώνυμους όρους από το `WordNet`.
3. Επαναλάβετε τα βήματα 3 έως 5 της Φάσης 1 για τα επαυξημένα με συνώνυμα ερωτήματά σας. Στην αναφορά σας συγκρίνετε τα αποτελέσματα της Φάσης 2 με της Φάσης 1. Υπάρχει κάποια βελτίωση; Προσπαθήσετε να αιτιολογήσετε τα αποτελέσματά σας είτε αυτά είναι θετικά είτε αρνητικά.

```
def get_synonyms(word):
    synonyms = set() for syn in wordnet.synsets(word):
    for lemma in syn.lemmas():
       synonyms.add(lemma.name().replace("_", " "))
    return list(synonyms)
```

**Κώδικας 1.0: Συνάρτηση για εύρεση συνωνύμων από το `WordNet`**

**Φάση 3 – Επέκταση ερωτήματος με συνώνυμα από το `word2vec` (15 μονάδες)**

Στην τρίτη φάση της εργασίας θα επεκτείνετε τα ερωτήματα της συλλογής `IR2025` με συνώνυμους όρους που θα αντλήσετε από το `word2vec`. Το `word2vec` είναι ένας αλγόριθμος που βασίζεται σε νευρωνικά δίκτυα πρόσθιας τροφοδότησης για τη μάθηση διανυσματικών αναπαραστάσεων των λέξεων που μπορούν να χρησιμοποιηθούν για την εύρεση λέξεων με παρόμοια σημασία ή λέξεων που εμφανίζονται σε παρόμοια περιβάλλοντα (contexts). Το μοντέλο εκτιμά την πιθανότητα να επιλεχθεί μία λέξη (output) με βάση το περιβάλλον της (input). Εξάγει τους κοντινότερους γείτονες μιας λέξης εξετάζοντας τα συμφραζόμενα, το περιβάλλον της λέξης και καθορίζει πότε δύο λέξεις είναι σημασιολογικά συναφείς (όταν εμφανίζονται σε ίδιο ή παρόμοιο περιβάλλον-context). Υπό το πρίσμα αυτό μπορούμε να χρησιμοποιήσουμε το μοντέλο για να ανακαλύψουμε συνώνυμες λέξεις.




1. Εκπαιδεύστε ένα μοντέλο `word2vec` χρησιμοποιώντας τη συλλογή `IR2025` ως είσοδο και τη βιβλιοθήκη `gensim`, η οποία παρέχει υλοποιημένα μοντέλα νευρωνικών δικτύων, όπως το `word2vec`, για την `python`. Η διαδικασία της παραγωγής του μοντέλου μπορεί να παραμετροποιηθεί ως προς την αρχιτεκτονική που θα χρησιμοποιηθεί, το μέγεθος παραθύρου ελέγχου και τον αριθμό των διαστάσεων. Οι διαθέσιμες αρχιτεκτονικές είναι οι `Skip-gram` και `CBOW` που θα αναλυθούν στο μάθημα. Το παράθυρο ελέγχου ορίζει τον αριθμό των λέξεων που θα λαμβάνονται υπόψη, πριν και μετά από την κάθε λέξη. Όσον αφορά τον αριθμό των διαστάσεων, αυτός ορίζει την πολυπλοκότητα και το μέγεθος του μοντέλου.
2. Επεκτείνετε τα ερωτήματα της `IR2025` με τους συνώνυμους όρους από το μοντέλο που κατασκευάσατε (δείτε κώδικα 2.0 για παράδειγμα).
3. Επαναλάβετε τα βήματα 3 έως 5 της Φάσης 1 για τα επαυξημένα με συνώνυμα ερωτήματά σας.

Στην αναφορά σας συγκρίνετε τα αποτελέσματα της Φάσης 3 με της Φάσης 1 και 2. Υπάρχει κάποια βελτίωση; Προσπαθήσετε να αιτιολογήσετε τα αποτελέσματά σας είτε αυτά είναι θετικά είτε αρνητικά.

```
Word2Vec model = Word2Vec(sentences, vector_size=100, window=5, min_count=1)

# Παράδειγμα εύρεσης λέξεων με παρόμοιο νόημα

word = "learning"

if word in model.wv:

   similar_words = model.wv.most_similar(word, topn=5)

   print(f"Top 5 λέξεις παρόμοιες με '{word}':")

   for w, score in similar_words:

      print(f"{w}: {score:.4f}")

   else:

      print(f"Η λέξη '{word}' δεν βρέθηκε στο λεξικό του μοντέλου.")
```

**Κώδικας 2.0: Python `word2vec` παράδειγμα**

**Φάση 4 BONUS (5 μονάδες)**

Η 4η φάση είναι προαιρετική. Θα ανακοινωθεί σύντομα!

| **ΠΡΟΣΟΧΗ ΣΤΙΣ ΦΑΣΕΙΣ 2, ΚΑΙ 3** |
|---|
| Η επέκταση των ερωτημάτων με συνώνυμους όρους πρέπει να γίνει με *προσοχή* τόσο στη Φάση 2 όσο και στη Φάση 3. Θα πρέπει να "ελέγξετε" τη διαδικασία διεύρυνσης έτσι ώστε να αποδίδονται συνώνυμα μόνο σε ορισμένους από τους όρους του ερωτήματος. Δεν έχουν όλοι οι όροι ενός ερωτήματος την ίδια σημασία. Για παράδειγμα, δεν ενδείκνυται να αποδώσετε συνώνυμους όρους στις τετριμμένες λέξεις. Τα αποτελέσματα μπορεί να χειροτερέψουν.<br><br>Στη Φάση 2 επιλέξτε τα συνώνυμα του `WordNet` για ορισμένα μόνο μέρη του λόγου. Για παράδειγμα, μπορείτε να χρησιμοποιήσετε συνώνυμα μόνο για τα ουσιαστικά ή τα επίθετα και όχι για τα ρήματα.<br><br>Στη Φάση 3 επιλέξτε τα συνώνυμα με τη μεγαλύτερη ομοιότητα ή με ομοιότητα μεγαλύτερη από ένα threshold.<br><br>Επίσης, μπορείτε να ορίσετε συνώνυμα μόνο για λέξεις του ερωτήματος που έχουν μεγάλο βάρος πχ. tfidf.<br><br>Δοκιμάστε τις προτεινόμενες λύσεις ή προτείνετε μια δική σας προσέγγιση στο πρόβλημα. |

**Υλοποίηση**

Η υλοποίηση της μηχανής αναζήτησης *προτείνεται* να πραγματοποιηθεί με χρήση `Python` και `ElasticSearch`. Η `ElasticSearch` είναι μία state-of-the-art μηχανή αναζήτησης με πολλές δυνατότητες. Μπορείτε να δοκιμάσετε μια άλλη μηχανή αναζήτησης (πχ. `Solr`) και άλλη γλώσσα προγραμματισμού (πχ. `java`), αλλά πιθανόν να έχετε περιορισμένη υποστήριξη από τη διδάσκουσα.

**Εργαλεία:** τα εργαλεία που θα χρειαστείτε μπορείτε να τα βρείτε παρακάτω

| | |
|---|---|
| `ElasticSearch 8.17.2` | https://www.elastic.co/elasticsearch<br>https://www.elastic.co/downloads/elasticsearch<br><br>https://www.elastic.co/guide/en/elasticsearch/reference/current/getting-started.html |
| `Python 3.x.y` | https://www.python.org/downloads/ |
| `trec_eval` | https://trec.nist.gov/trec_eval/ (διαθέσιμο στο eclass) |
| `WordNet` | https://wordnet.princeton.edu |

**Συλλογή `IR2025`**

Επιλέξτε συλλογή ως εξής: Αν το άθροισμα των αριθμών μητρώου σας τελειώνει σε ζυγό αριθμό επιλέξτε τη συλλογή scidocs. Αν το άθροισμα των αριθμών μητρώου σας τελειώνει σε περιττό αριθμό επιλέξτε τη συλλογή trec_covid. Οι συλλογές είναι κατάλληλες για την αξιολόγηση του συστήματός σας. Είναι σε μορφή .jsonl.

Κατεβάστε τη συλλογή σας από τα παρακάτω links:

Συλλογή scidocs.zip

Συλλογή trec-covid.zip

**Βιβλιογραφία - Πηγές:**

Mikolov, T., Corrado, G., Chen, K., & Dean, J. (2013). Efficient Estimation of Word Representations in Vector Space. Proceedings of the International Conference on Learning Representations (ICLR 2013), 1–12.

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Distributed Representations of Words and Phrases and their Compositionality. NIPS, 1–9.

Η προγραμματιστική εργασία είναι ομαδική (ομάδες 2 φοιτητών), είναι υποχρεωτική, πιάνει το 40% του τελικού βαθμού σας (+5% bonus), και θα προσμετρηθεί στον τελικό βαθμό σας αν ο βαθμός γραπτής εξέτασης είναι μεγαλύτερος ή ίσος του 4. Στο τέλος του εξαμήνου θα πραγματοποιηθεί υποχρεωτική ατομική προφορική εξέταση, από την οποία θα εξαρτηθεί ο τελικός βαθμός κάθε φοιτητή στην εργασία.

**Ημερομηνίες υποβολής φάσεων εργασίας (ενδέχεται να τροποποιηθούν)**

Φάση 1: 18 Μαΐου 2025

Φάση 2: 25 Μαΐου 2025

Φάση 3: ίσως ημ/νία εξέτασης μαθήματος

Φάση 4: ίσως ημ/νία εξέτασης μαθήματος