ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

Τμήμα Πληροφορικής

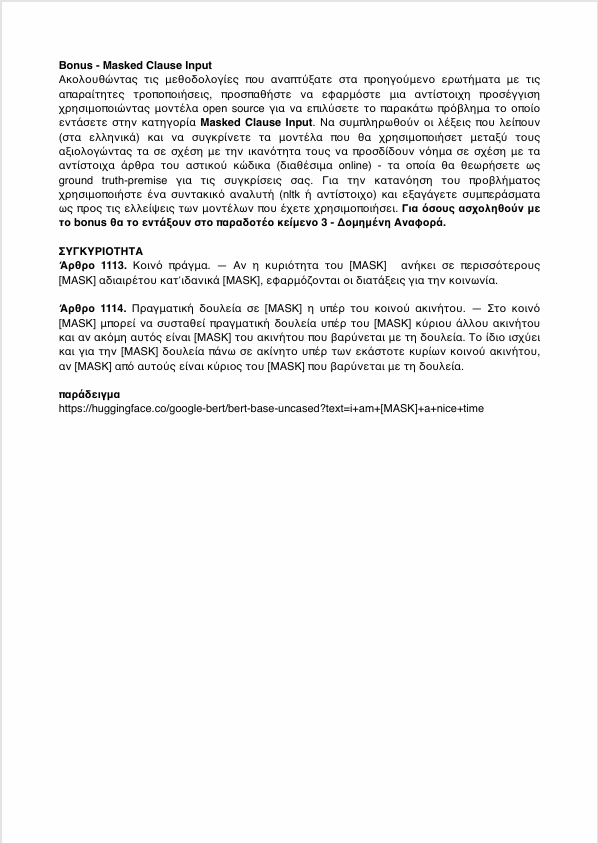
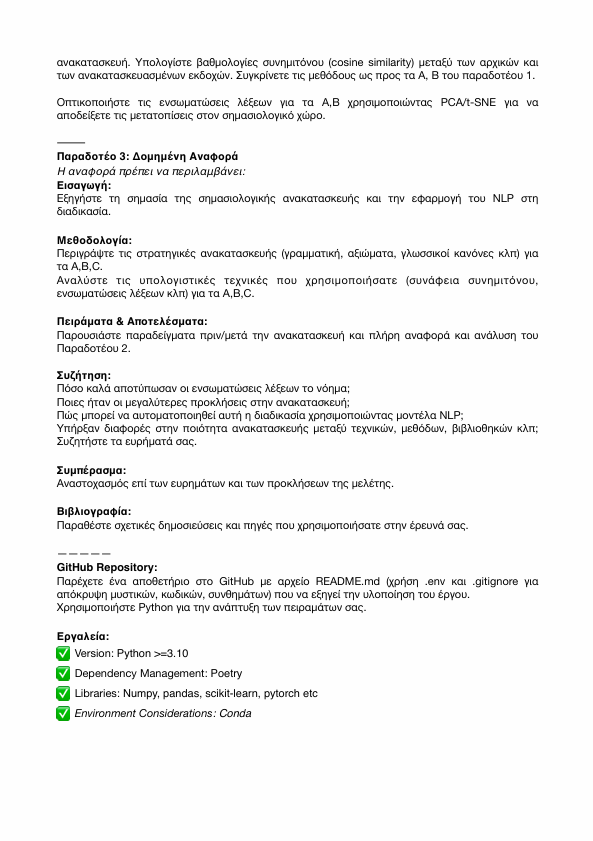
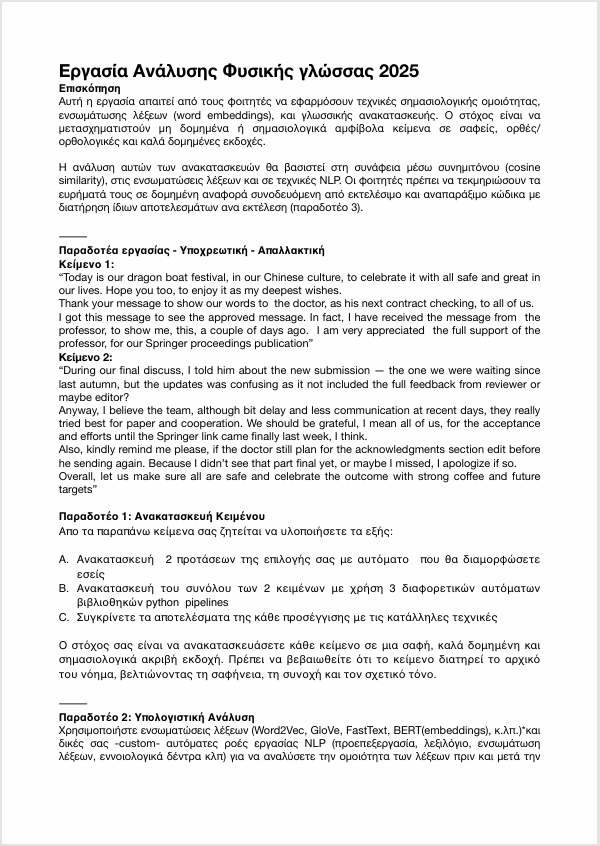


Εργασία Μαθήματος ***Επεξεργασία Φυσικής Γλώσσας***

***Παραδοτέο 1***

|  |  |
| --- | --- |
| ***Αριθμός εργασίας – Τίτλος εργασίας*** | ***Απαλλακτική*** |
| Όνομα φοιτητή | ΜΑΡΙΟΣ ΚΥΡΟΓΛΟΥ |
| Αρ. Μητρώου | Π21080 |
| Ημερομηνία παράδοσης |  |

**Εκφώνηση εργασίας**



ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

[1 Εισαγωγή 6](#_Toc198156296)

[2 Επίδειξη της λύσης 7](#_Toc198156297)

[3 Βιβλιογραφικές Πηγές 12](#_Toc198156298)

1. Εισαγωγή

Αρχικά, μετά από προσεκτική μελέτη της εκφώνησης, αποφάσισα να υλοποιήσω την εργασία σε γλώσσα προγραμματισμού Python και ως εργαλείο ανάπτυξης του project επέλεξα την πιο νέα και πληρέστερη έκδοση του Conda. Επιπλέον χρησιμοποιήθηκαν βιβλιοθήκες: Numpy, pandas, scikit-learn, pytorch. Τέλος αφού καταγράφηκαν οι βασικές απαιτήσεις του θέματος, δημιουργήθηκαν αναφορές σχετικά με την σύγκριση των αποτελεσμάτων της κάθε προσέγγισης με τις κατάλληλες τεχνικές όσον αφορά το **παραδοτέο 1**.

1. Επίδειξη της λύσης

**Περιγραφή των προσεγγίσεων**

* **Μέρος Α**

Ανακατασκευή 2 προτάσεων της επιλογής σας με αυτόματο που θα διαμορφώσετε εσείς

* **Μέρος Β**

Ανακατασκευή του συνόλου των 2 κειμένων με χρήση 3 διαφορετικών αυτόματων βιβλιοθηκών python pipelines.

**Μοντέλο**

* **Μέρος Α**

**Word2Vec** εκπαιδευμένο επιτόπου σε κάθε κείμενο.

* **Μέρος Β**

– **NLTK**: W2V τοπικό

– **spaCy**: προεκπαιδευμένο en\_core\_web\_sm vector

– **TextBlob**: συνδυασμός spaCy vectors ανά λέξη

**Ανάλυση Συγκρίσης Τεχνικών**

1. **Εκπαίδευση μοντέλων ενσωμάτωσης**
   * **thema1A:** Το **Word2Vec** εκπαιδεύεται επί τόπου σε πολύ μικρό corpus (λίγες δεκάδες προτάσεις). Αυτό οδηγεί σε embeddings που αντανακλούν μόνο το συγκεκριμένο κείμενο, αλλά κινδυνεύει από υπερεκπαίδευση και φτωχή γενίκευση.
   * **thema1B:** Χρήση προεκπαιδευμένου **spaCy vector** (πολύ μεγαλύτερο corpus) εξασφαλίζει πιο σταθερές και γενικεύσιμες αναπαραστάσεις, ενώ το τοπικό W2V στο NLTK pipeline παραμένει ευαίσθητο στο domain.
2. **Βάθος παραφράσεων**
   * **thema1A:** Απλό word‐level paraphrasing, χωρίς συντακτική ή σημασιολογική αναδιάταξη.
   * **thema1B:** Διαφορετικά επίπεδα:
     + **NLTK:** αντίστοιχο με A
     + **spaCy:** λεκτικό με διατήρηση POS, πιο επιλεκτικό
     + **TextBlob:** βασισμένο σε POS‐tagging του TextBlob, μπορεί να αλλάξει λέξεις πιο “φυσικά”
3. **Κλίμακα αξιολόγησης**
   * **thema1A:** Μετρά ομοιότητα ανά ζεύγος προτάσεων – ιδανικό για λεπτομερή έλεγχο ποιότητας παραφράσεων προτάσεων.
   * **thema1B:** Μετρά ομοιότητα ολόκληρου του κειμένου – χρήσιμο για συνολική διατήρηση νοήματος, αλλά χάνει λεπτομέρειες ανά πρόταση.

**Ερμηνεία Αποτελεσμάτων**

* **thema1A**: Τυπικές τιμές cosine similarity ~0.7–0.9 ανά πρόταση δείχνουν υψηλή διατήρηση σημασίας παρά τις τυχαίες αντικαταστάσεις. Ωστόσο, μικρές αποκλίσεις μπορεί να οφείλονται σε ονόματα, αριθμούς ή σπάνιες λέξεις που δεν έχουν συνώνυμο .
* **thema1B**:
  + **NLTK‐W2V**: συχνά η χαμηλότερη ομοιότητα (π.χ. ~0.65), λόγω συσσώρευσης σφαλμάτων σε μακροσκελείς παραφράσεις.
  + **spaCy**: η υψηλότερη ομοιότητα (π.χ. ~0.85–0.9), χάρη σε πιο πλούσια, προεκπαιδευμένα vectors.
  + **TextBlob**: ενδιάμεση τιμή (~0.75–0.8), αλλά με μεγαλύτερη ποικιλία λέξεων, καθώς βασίζεται σε POS και μπορεί να επιλέγει λιγότερο συχνά synonyms .

**Πλεονεκτήματα & Μειονεκτήματα**

* **Πλεονεκτήματα**

**Θέμα 1 – Μέρος Α (thema1A)**

* Εστίαση σε κρίσιμες προτάσεις.
* Γρήγορη εκτέλεση σε μικρό corpus.

**Θέμα 1 – Μέρος Β (thema1Β)**

* Συγκρίσιμα αποτελέσματα τριών pipelines.
* Ολοκληρωμένη ανάλυση ολόκληρου κειμένου.
* Χρήση ισχυρών προεκπαιδευμένων μοντέλων (spaCy).
* **Μειονεκτήματα**

**Θέμα 1 – Μέρος Α (thema1A)**

* Ευαίσθητη σε τυχαία επιλογή.
* Μη γενικεύσιμη σε άλλο κείμενο.

**Θέμα 1 – Μέρος Β (thema1Β)**

* Μεγάλος υπολογιστικός φόρτος.
* Αθροιστική φθορά ποιότητας σε μακροσκελείς παραφράσεις.
* Πιθανή ασυνέπεια μεταξύ pipeline.

**Συμπεράσματα**

Από την ανάλυση των δύο μεθοδολογιών παραφράσεων φαίνεται καθαρά ότι η **thema1A.py** προσφέρει μία ταχεία και εστιασμένη λύση, ιδανική για τη γρήγορη διαχείριση και ποιοτικό έλεγχο περιορισμένου αριθμού κρίσιμων προτάσεων, αλλά χωρίς την ικανότητα να διατηρήσει τη συνοχή σε μεγαλύτερα κείμενα ή να γενικεύσει σε διαφορετικά πεδία. Αντιθέτως, η **thema1B.py**, υποστηριζόμενη από προεκπαιδευμένα embeddings υψηλής ποιότητας (ιδιαίτερα μέσω του spaCy pipeline), καταδεικνύει σημαντικά πλεονεκτήματα σε ό,τι αφορά τη διατήρηση του συνολικού νοήματος και την αξιοπιστία των αποτελεσμάτων, παρά τον αυξημένο υπολογιστικό φόρτο. Η διαφοροποίηση των pipelines (NLTK, spaCy, TextBlob) στην **thema1B.py** προσφέρει ευελιξία και ποικιλία επιλογών, με το spaCy να επιτυγχάνει τις υψηλότερες τιμές ομοιότητας cosine, ενώ το NLTK–W2V να υπολείπεται εν μέρει λόγω του τοπικού περιορισμένου corpus. Συνολικά, για εφαρμογές μικρής κλίμακας που απαιτούν ταχύτητα και απλότητα, η θεματική της **thema1A.py** είναι κατάλληλη, ενώ για εκτενείς αναλύσεις όπου προέχει η ακεραιότητα και η συνέπεια του κειμένου, η πολυπρισματική προσέγγιση της **thema1B.py** υπερέχει.