

ΑΠΑΛΑΚΤΙΚΗ ΕΡΓΑΣΙΑ ΣΤΟ ΜΑΘΗΜΑ

ΕΠΕΞΕΡΓΑΣΙΑ ΦΥΣΙΚΗΣ ΓΛΩΣΣΑΣ

ΕΚΠΟΝΗΘΗΚΕ ΑΠΟ ΤΟΝ ΕΚΠΑΙΔΕΥΟΜΕΝΟ

ΟΔΥΣΣΕΑ ΤΣΟΥΡΓΙΑΝΝΗ Π22184

Εισαγωγή

Στη σύγχρονη υπολογιστική γλωσσολογία, η σημασιολογική ανακατασκευή αποτελεί έναν από τους βασικότερους στόχους της Επεξεργασίας Φυσικής Γλώσσας (Natural Language Processing – NLP). Πρόκειται για τη διαδικασία με την οποία επιδιώκουμε να αναπαραστήσουμε τη σημασία λέξεων, φράσεων ή κειμένων με τέτοιο τρόπο ώστε να μπορούμε να τις συγκρίνουμε, να εντοπίζουμε αλλαγές ή να αντλούμε πληροφορία με βάση το νόημα και όχι απλώς το λεξιλόγιο. Μια από τις πιο διαδεδομένες τεχνικές για την επίτευξη αυτής της αναπαράστασης είναι η χρήση σημασιολογικών διανυσμάτων λέξεων (word embeddings), όπως αυτά που παράγονται από προ-εκπαιδευμένα μοντέλα, π.χ. το Word2Vec της Google. Τα διανύσματα αυτά τοποθετούν κάθε λέξη σε έναν πολυδιάστατο χώρο, όπου η γεωμετρική εγγύτητα μεταξύ των λέξεων αντανακλά τη σημασιολογική τους ομοιότητα.

Στο πλαίσιο αυτής της εργασίας ζητείται η παράφραση 2 κειμένων. Στην υλοποίηση, εφαρμόζονται διάφορα μοντέλα παραφράσεων καθώς και τεχνική μείωσης διαστάσεων μέσω PCA (Principal Component Analysis), ώστε τα διανύσματα των ενσωματωμένων λέξεων να προβληθούν σε δισδιάστατο χώρο. Η οπτικοποίηση αυτή επιτρέπει την παρακολούθηση αλλαγών στη σημασία και τη χρήση λέξεων ανάμεσα σε δύο κείμενα

(αρχικό και τελικό), αποκαλύπτοντας πιθανούς μετασχηματισμούς ή σημασιολογικές αποκλίσεις.

Μεθοδολογία

Παραδοτέο 1^ο

A)

Στο πρώτο μέρος του πρώτου παραδοτέου της εργασίας μας ζητείται η δημιουργία ενός δικού μας αυτόματου για την ανακατασκευή 2 προτάσεων της επιλογής μας από τα 2 κείμενα. Επέλεξα τις προτάσεις : "Hope you too, to enjoy it as my deepest wishes."→κείμενο 1, " Overall, let us make sure all are safe and celebrate the outcome with strong coffee and future targets."→κείμενο 2.

Η πρώτη μας κίνηση είναι να κάνουμε import την random και να εκχωρήσουμε τις προτάσεις σε μία λίστα "sentences", έπειτα θα επιλέξουμε τις κομβικές για την σημασιολογία των προτάσεων λέξεις και σε ένα λεξικό "synonyma" θα αναθέσουμε για κάθε μια μερικά συνώνυμα της. Θα ξεκινήσουμε την υλοποίηση της συνάρτησης ανακατασκευής που θα δέχεται ένα argument , τις προτάσεις που έχουμε επιλέξει. Αρχικοποιούμε έναν πίνακα new_sentences[] για την αποθήκευση μετέπειτα των νέων προτάσεων.

Συνεχίζοντας, για κάθε s στο sentences στην μεταβλητή words αποθηκεύουμε το s.split() δηλαδή τις λέξεις της πρότασης και αρχικοποιούμε έναν πίνακα new_sen[] που θα κρατάει μία ανακατασκευασμένη πρόταση τη φορά. Για κάθε word* στα words θα φτιάξουμε τα υποψήφια key_words του λεξικού μας κάνοντας κάθε λέξη strip από σημεία στίξης και τα γράμματα της μικρά. Αν το key_word υπάρχει στα συνώνυμα διαλέγουμε μια τυχαία τιμή από τα values και την αποθηκεύουμε σε μία μεταβλητή new_word και παράλληλα εάν το πρώτο γράμμα του word* ήταν κεφαλαίο κάνουμε new_word.capitalize(), αποθηκεύοντας στον πίνακα new_sen. Αν το key_word δεν υπάρχει στα συνώνυμα την προσθέτουμε αυτούσια στον new_sen μετά την διαδικασία των ελέγχων μετατρέπουμε την πρόταση new_sen σε string και προσθέτουμε στον new_sentences επιστρέφουμε εκτός των διαδικασιών , το αποτέλεσμα.

Σε μια μεταβλητή rec_sen καλούμε την cnstr(sentences) και εκτυπώνουμε την rec_sen παίρνοντας το αποτέλεσμα με βάση τις random επιλογές που έγιναν.

```
['Wish you too, to like it as my best prayers', 'Altogether let us assure all are secure and enjoy the result with rich coffee and next goals']  
Press any key to continue . . . ■
```

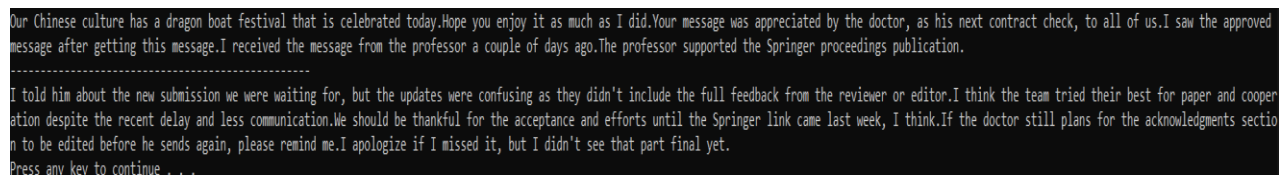
Εικόνα 1: Εκτύπωση 1^Α

B)

Στο δεύτερο ερώτημα μας ζητείται να χρησιμοποιήσουμε 3 αυτόματες βιβλιοθήκες pipeline της pythοn για να παραφράσουμε τα κείμενα μας. Πρωτού ξεκινήσουμε θα κάνουμε import την pipeline `from transformers import pipeline` και θα χωρίσουμε τα κείμενα σε προτάσεις προσθέτοντας στους πίνακες `keimeno1` και `keimeno2`. Όλα τα μοντέλα που θα χρησιμοποιήσουμε λειτουργούν καλύτερα σε μικρά κομμάτια κειμένου, έτσι επιτυγχάνουμε να είναι η αισθητή η επίδραση του μοντέλου πάνω στο κείμενο.

1^η Παράφραση → Pegasus)

Αρχικά δημιουργούμε ένα pipeline και του φορτώνουμε το μοντέλο αυτό το αποθηκεύουμε σε μία μεταβλητή `paraphraser`. Δημιουργούμε τη συνάρτηση `pegasus` που δέχεται το `text` δηλαδή το κείμενο προς παράφραση και τον αριθμό παραλλαγών που θα επιστρέψει. Εντός της συνάρτησης σε μία μεταβλητή `results` αποθηκεύουμε το λεξικό με τις παραλλαγές του `text` και επιστρέφουμε μόνο τις παραλλαγές. Αρχικοποιούμε τα `string tel_keim1_1, tel_keim2_1` που θα φιλοξενήσουν τα τελικά παραφρασμένα κείμενα και για κάθε `s` στο κείμενο (1 και 2 ξεχωριστά), παίρνουμε το αποτέλεσμα της `pegasus` και εκχωρούμε μόνο το πρώτο αποτέλεσμα της επιστροφής στο `string` του αντίστοιχου κειμένου που κάνουμε την παράφραση. Σημείωση : Η επιστροφή της `Pegasus` είναι μια λίστα από παραφράσεις εμείς επιλέγουμε 1 από αυτή πχ την πρώτη για να προσθέσουμε στο τελικό κείμενο. Τέλος εκτυπώνουμε τα `tel_keim1_1, tel_keim2_1` (θα μπορούσαμε να ρυθμίσουμε την επιστροφή παραφράσεων σε 1 χωρίς να χρειάζεται να προσθέτουμε κάθε φορά το πρώτο στοιχείο και να μειώσουμε και τον χρόνο που απαιτεί το πρόγραμμα για να εκτελεστεί. Ο λόγος που δεν έγινε αυτό, είναι η περίπτωση που επιθυμείτε κατά την εξέταση να δείτε και τις υπόλοιπες παραφράσεις) . [5]



Our Chinese culture has a dragon boat festival that is celebrated today.Hope you enjoy it as much as I did.Your message was appreciated by the doctor, as his next contract check, to all of us.I saw the approved message after getting this message.I received the message from the professor a couple of days ago.The professor supported the Springer proceedings publication.

.....

I told him about the new submission we were waiting for, but the updates were confusing as they didn't include the full feedback from the reviewer or editor.I think the team tried their best for paper and cooperation despite the recent delay and less communication.We should be thankful for the acceptance and efforts until the Springer link came last week, I think.If the doctor still plans for the acknowledgments section to be edited before he sends again, please remind me.I apologize if I missed it, but I didn't see that part final yet.

Press any key to continue . . .

Εικόνα 2: Εκτύπωση B^{10}

2^η Παράφραση →Bart-Large)

Όπως και πριν δημιουργούμε το Pipeline και αρχικοποιούμε ξανά αλλά αυτή τη φορά ως `tel_keim1_2, tel_keim2_2`. Με 2 loops δημιουργούμε για κάθε `s` στα κείμενα το αποτέλεσμα του `paraphraser`, η μόνη διαφορά είναι πως η επιστροφή δεν είναι μια λίστα παραφράσεων αλλά μία λίστα λεξικών, άρα για να αποθηκεύσουμε την πρώτη παράφραση από το πρώτο λεξικό γράφουμε: `tel_keim1+= result[0]['generated_text']+" "`. Στο τέλος γίνεται η εκτύπωση των `tel_keim1_2` και `tel_keim2_2`. [1]

Today is our dragon boat festival, in our Chinese culture, to celebrate it with all safe and great in our lives. Today is our Dragon Boat Festival. To celebrate it, go to your local dragon boat club. Go to dragonboatclub.com to find out how to go. "I hope you too, to enjoy it as my deepest wishes. Hope you too, to enjoy the rest of my life," he said. "I love you all, and I hope you will love me too," he added. "It is my deepest wish that you will all enjoy it." "Thank your message to show our words to the doctor, as his next contract checking, to all of us. Thank your message, as our next contract check, to be sure he is doing what he should be doing," the group said in a statement. "We are all in this together," they added. I got this message to see the approved message. I got this messages to see the approved messages. I go to it to say that I had been sent a message to the wrong person. I was sent the wrong message to a person who had already sent me a message. I am happy to clarify this. "I have received the message from the professor, to show me, this, a couple of days ago. In fact, I have received a message from the professor," he said. "I have been shown this, and I have been told that it is a good thing," he added. "I am very appreciated the full support of the professor, for our Springer proceedings publication," says the author. "I am also very grateful to the professor for his full support for the publication of the Springer proceedings," he adds. "It was a great honour to be able to work on this project," adds the author of the book.

"I told him about the new submission – the one we were waiting since last autumn, but the updates were confusing as it not included the full feedback from reviewer or maybe editor?" he said. "During our final discuss, I told him of the new Submission – the one we were waiting for since last autumn, and the updates were confusing as they not included full feedback" "I believe the team, although bit delay and less communication at recent days, they really tried best for paper and cooperation. Anyway, I believe they really try best," he said. "I believe that they really did," he added. "They really did try best" We should be grateful, I mean all of us, for the acceptance and efforts until the Springer link came finally last week, I think. We should be grateful, for all the acceptance and efforts until this week, I think. I think we should be grateful. The doctor still plan for the acknowledgments section edit before he sending again. Also, kindly remind me please, if the doctor still plans on editing the acknowledgments section before he sends again, that I should edit it again before sending. I'm not a doctor, I'm just a writer. I didn't see that part final yet, or maybe I missed, I apologize if so. Overall, let us make sure all are safe and celebrate the outcome with strong coffee and future targets. Because I didn't see that final part yet, I apologize if I missed.

Press any key to continue . . .

Εικόνα 3: Εκτύπωση B^{20}

3^η Παράφραση → Ramsrigouthamg)

Ακριβώς την ίδια διαδικασία ακολουθούμε και εδώ αρχικοποιώντας τα tel_keim1_3, tel_keim2_3, φορτώνοντας το paraphraser και παραφράζοντας τα κείμενα για την κατασκευή και την εκτύπωση των tel_keim1_3, tel_keim2_3. [2]

Today is our dragon boat festival, in our Chinese culture, to celebrate it with all safe and great in our lives. I hope you too, to enjoy it as my deepest wishes. Thank you for your message to show our words to the doctor, as his next contract checking, to all of us. I got this message to see the approved message. In fact, I have received the message from the professor, to show me, this, a couple of days ago. I am very appreciated the full support of the professor, for our Springer proceedings publication.

During our final discuss, I told him about the new submission – the one we were waiting since last autumn, but the updates were confusing as it not included the full feedback from reviewer or maybe editor? Anyway, I believe the team, although bit delay and less communication at recent days, they really tried best for paper and cooperation. We should be grateful, I mean all of us, for the acceptance and efforts until the Springer link came finally last week, I think. Also, kindly remind me please, if the doctor still plan for the acknowledgments section edit before he sending again. Because I didn't see that part final yet, or maybe I missed, I apologize if so. Overall, let us make sure all are safe and celebrate the outcome with strong coffee and future targets.

Press any key to continue . . .

Εικόνα 3: Εκτύπωση B^{30}

Γ)

Μετά τις τρεις εκτυπώσεις μπορούμε να σχολιάσουμε ότι το Pegasus μοντέλο αποδίδει μία πολύ συμπαγή και σημασιολογικά πιο κοντινή στα original κείμενα, παράφραση. Στο BART- Large, παρατηρούμε πολλές επαναλήψεις φράσεων και πολλές προσθήκες νέου κειμένου που σημασιολογικά δεν συμπίπτουν με τα αρχικά κείμενα ή και από μόνες τους πολλές φορές δεν βγάζουν κάποιο ουσιαστικό νόημα. Παρόλα αυτά, καταφέρνει να κρατήσει μία κεντρική ιδέα, αλλά σε καμία περίπτωση δεν πλησιάζει την σαφήνεια του Pegasus. Τέλος, το μοντέλο Ramsrigouthamg καταφέρνει να αποδώσει την καλύτερη απ όλες παράφραση, σημασιολογικά δεν ξεφεύγει από το αρχικό κείμενο, και εάν εκτυπώσουμε όλες τις παραλλαγές που επιστρέφονται θα δούμε ότι θα υπάρχουν ελάχιστα σημεία που οι φράσεις που αναπαράγει, είναι αφηρημένου νοήματος.

Αξίζει επίσης να σημειώσουμε ότι το πρώτο κείμενο είναι σημασιολογικά πιο χαμένο από το δεύτερο, με αποτέλεσμα τα περισσότερα λάθη από τα τρία μοντέλα, να παρατηρούνται σε αυτό. Αρχικός στόχος μας ήταν η παραγωγή δύο καλά δομημένων και σημασιολογικά κοντινών παραλλαγών των δύο κειμένων, πράγμα που καταφέρνει καλύτερα από όλα τα μοντέλα το μοντέλο Ramsrigouthamg. Ενώ τελειώνοντας, αξίζει να παρατηρήσουμε ότι κανένα από τα δύο αρχικά κείμενα δεν έχει ξεκάθαρη σημασιολογία και κεντρικό θέμα, πράγμα που δεν ευνοεί κανένα από τα μοντέλα που χρησιμοποιήθηκαν, γι' αυτό παρατηρούμε και σοβαρά λάθη σε μερικά σημεία.

Σημείωση για παραδοτέο 1^ο :

Καθώς τα δύο κείμενα που έχουν δοθεί θεωρούνται δεδομένα και είναι ίδια για όλους τους εκπαιδευόμενους, δεν εκτυπώνονται κατά την εκτέλεση του προγράμματος. Ο λόγος που αποφασίστηκε να μην εκτυπωθούν - κάθε φορά που εκτυπώνεται μια παράφραση τους, αποτελεί το γεγονός ότι πολλές φορές στον κώδικα απαιτείται η εκτύπωση παραφρασμένου κειμένου. Θεωρήθηκε λοιπόν σημαντικό, η εκτύπωση να είναι ξεκάθαρη ως προς τα αποτελέσματα χωρίς να δημιουργείται σύγχυση με παρεμβαλλόμενη αναπαράσταση δεδομένων της εκφώνησης.

Παραδοτέο 2^ο

Ξεκινώντας την τεκμηρίωση του παραδοτέου 2 και αποσκοπώντας να μετρήσουμε την ομοιότητα των παραφρασμένων κειμένων με τα αρχικά θα αρχικοποιήσουμε στις `keimeno_1ολοκληρο`, `keimeno2_ολοκληρο` ολόκληρα τα κείμενα μας. Για την έτοιμη αυτόματη διαδικασία θα επιλέξουμε το μοντέλο Sentence – Bert και θα το φορτώσουμε στην μεταβλητή `model` ενώ για την κάλυψη του custom NLP που απαιτείται θα υλοποιήσουμε μία δική μας συνάρτηση. Η συνάρτηση `clean_text`, δέχεται ένα `text string` και αποδίδει με την βοήθεια της `re`, το μήνυμα χωρίς ειδικούς χαρακτήρες, με μικρά όλα τα γράμματα καθαρίζοντας μερικούς παράγοντες που ενδέχεται να μπερδέψουν το μοντέλο.

Σειρά έχει η υλοποίηση της συνάρτησης υπολογισμού Cosine Similarity [4] χρησιμοποιώντας την συνάρτηση `util.cos_sim` : `from sentence_transformers import SentenceTransformer, util`. Δέχεται σαν arguments το αρχικό και το παραφρασμένο κείμενο, καθώς και ένα `string` για να υπάρχει δυναμικό output. Κωδικοποιούμε τα δύο κείμενα με την συνάρτηση `encode.model()` και υπολογίζουμε το Cosine Similarity των δύο κωδικοποιήσεων συνοδευόμενο από ένα dynamic output για την ανακοίνωση των αποτελεσμάτων.

Για την εκτύπωση των αποτελεσμάτων περνάμε όλα τα τελικά και τα αρχικά κείμενα από την `clean_text` και καλούμε συνολικά 6 φορές την συνάρτηση υπολογισμού, 2 για κάθε μοντέλο παράφρασης. Παραδείγματος χάρη, για την παράφραση του πρώτου κειμένου με Pegasus θα καλέσουμε την `cos_υπολογισμος` και θα περάσουμε το πρώτο αρχικό κείμενο, το πρώτο παραφρασμένο με Pegasus και το μήνυμα που συγκεκριμενοποιεί το αποτέλεσμα.

Παρακάτω (εικόνα 4) βλέπουμε τα αποτελέσματα cosine similarity για όλες τις παραφράσεις που πραγματοποιήθηκαν και αξίζει να σχολιάσουμε ότι : Η εικόνα 4 επιβεβαιώνει ακριβώς αυτό που παρατηρήθηκε και σχολιάστηκε σε προηγούμενο ερώτημα για τα δύο κείμενα της εκφώνησης. Η παράφραση με το μοντέλο Pegasus αποδίδει μια αρκετά καλής ποιότητας παράφραση με τη βαθμολογία του συνημίτονου να είναι κοντά στο 1. Τεκμηριώνεται επίσης, το γεγονός ότι η παράφραση με τη χρήση του μοντέλου BART

- Large είναι η χειρότερη από τα τρία μοντέλα, ενώ η παράφραση Ramsrigouthamg αποδίδει τις καλύτερες με βαθμό σχεδόν άριστο. Τέλος, ακόμα κάτι που μπορούμε να σημειώσουμε είναι ότι σε όλες τις παραφράσεις παρατηρείται στο πρώτο κείμενο χαμηλότερη βαθμολογία από ότι στο δεύτερο. Αυτό τεκμηριώνει ακριβώς αυτό που είχαμε σχολιάσει προηγουμένως, ότι το πρώτο κείμενο είναι νοηματικά πιο χαοτικό από το δεύτερο.

```
Cosine similarity πρώτης πρότασης δικού μου αυτόματου: 0.6361
Cosine similarity δεύτερης πρότασης δικού μου αυτόματου: 0.7676
-----
Cosine similarity πρώτου κειμένου Pegasus: 0.8930
Cosine similarity δεύτερου κειμένου Pegasus: 0.9439
-----
Cosine similarity πρώτου κειμένου BART -large: 0.7766
Cosine similarity δεύτερου κειμένου BART -large: 0.9096
-----
Cosine similarity πρώτου κειμένου Ramsrigouthamg: 0.9888
Cosine similarity δεύτερου κειμένου Ramsrigouthamg: 0.9923
Press any key to continue . . .
```

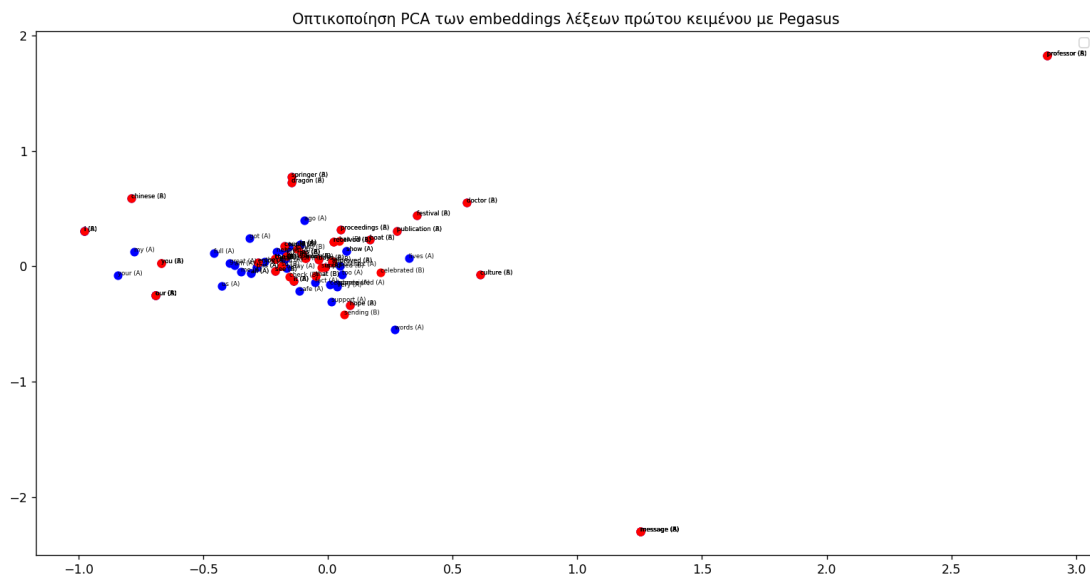
Εικόνα 4: Εκτύπωση Cosine Similarity

Για την ολοκλήρωση των απαιτήσεων της εργασίας απαιτείται η τεκμηρίωση της οπτικοποίηση των ενσωματώσεων στα 2 κείμενα με τα 3 μοντέλα. Πριν υλοποιήσουμε την συνάρτηση θα φορτώσουμε το μοντέλο word embendings από το Google News Word2Vec με την βοήθεια της βιβλιοθήκης `import gensim.downloader as api` θα χρειαστούμε την `import matplotlib.pyplot as plt` και `import numpy as np` για την οπτικοποίηση και την `from sklearn.decomposition import PCA` για την χρήση της κλάσης PCA [3].

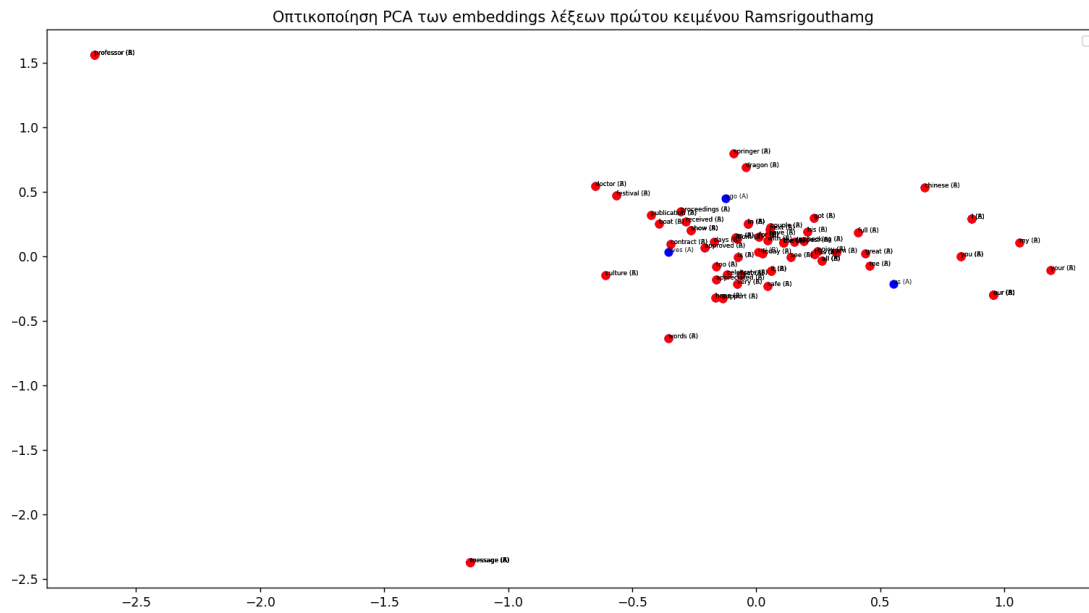
Η συνάρτηση που θα οπτικοποιεί τις ενσωματώσεις δέχεται το αρχικό κείμενο , το τελικό κείμενο , το μοντέλο που χρησιμοποιείται ("`word2vec-google-news-300`") και ένα label για να υπάρχει δυναμική εκτύπωση για πιο κατανοητά αποτελέσματα. Στο εσωτερικό της συνάρτησης για το αρχικό και το τελικό κείμενο split- άρουμε τις λέξεις και κρατάμε σε δύο μεταβλητές όσες λέξεις βρίσκονται μέσα στο μοντέλο (δηλ. έχουν διάνυσμα με το μοντέλο αυτό) για το αρχικό και το τελικό κείμενο.

Συνεχίζοντας για κάθε λέξη στο `arhiko_words` η `teliko_words` αποθηκεύουμε το διάνυσμα της στην αντίστοιχη μεταβλητή. Εν συνεχεία με την χρήση της `np.vstack` ενώνουμε τους 2 πίνακες με τα διανύσματα και δημιουργούμε ένα αντικείμενο 2D της κλάσης `pca` το οποίο εκπαιδεύουμε με τα συνολικά διανύσματα με την βοήθεια της `pca.fit_transform()`. Η εκπαίδευση αποδίδει έναν `numpy array` με συντεταγμένες 2D – διανύσματα τα οποία τα αποθηκεύουμε με το όνομα `reduced`. Σε `for-loop` με χρήση της

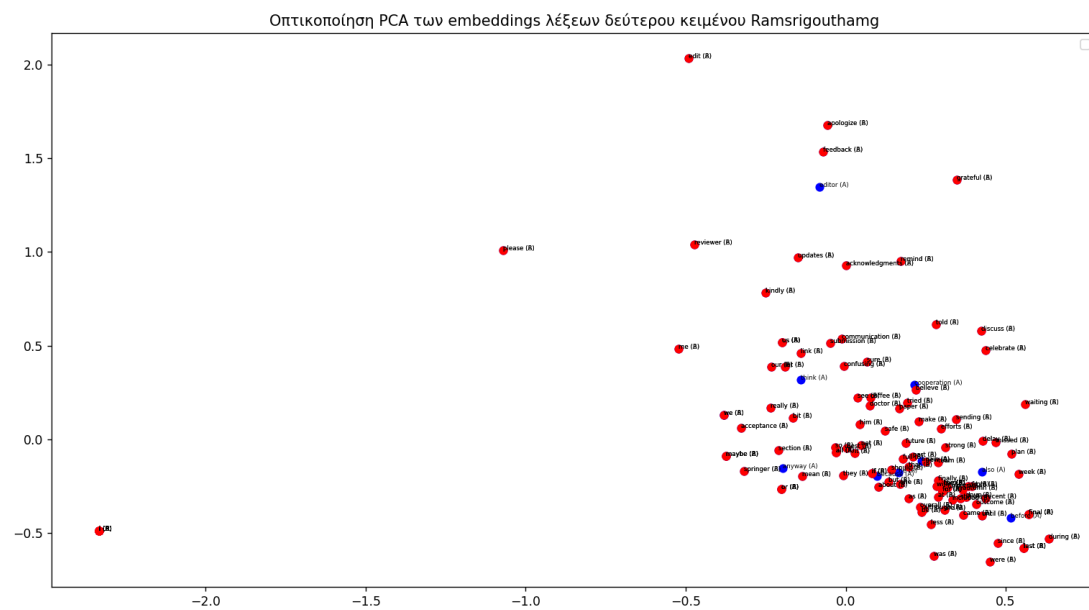
Για να προχωρήσουμε στο κείμενο 2 πρέπει να σκεφτούμε ότι \rightarrow Ο πίνακας `Reduced` περιέχει πρώτα τα διανύσματα για τις λέξεις του πρώτου κειμένου και μετά για του δεύτερου κειμένου. Ο μετρητής `i`, με το τέλος της επανάληψης αυτής, θα έχει την τιμή `len(archiko_words) - 1` οπότε στο επόμενο `loop` για `j, word` στα `teliko_words` οι τοποθετήσεις θα ξεκινάνε από την θέση `i+1+j`. Διαφοροποιούμε επίσης το χρώμα σε κόκκινο και το συμπληρωματικό `text` σε (B) αντί (A) και τέλος προσθέτουμε τίτλο στον καμβά χρησιμοποιώντας το `label` και τον εμφανίζουμε. Για την εκτύπωση των αποτελεσμάτων, σε κάθε παράφραση, για κάθε κείμενο, θα καλέσουμε την συνάρτηση `pca` δίνοντας το αρχικό ολόκληρο κείμενο, το τελικό κείμενο, το μοντέλο, δηλαδή το `model2` που φορτώσαμε στην αρχή, και το `label` το οποίο θα βρίσκεται στον τίτλο. Συμπεριλαμβανομένου και του ερωτήματος A από το παρεδοτέο 1, τα διαγράμματα τα οποία θα εμφανιστούν κατά την εκτέλεση είναι 8.



Εικόνα 5: Διάγραμμα οπτικοποίησης ενσωματώσεων



Εικόνα 6: Διάγραμμα οπτικοποίησης ενσωματώσεων



Εικόνα 7: Διάγραμμα οπτικοποίησης ενσωματώσεων

Συζήτηση

Με την ολοκλήρωση της τεκμηρίωσης του κώδικα και πλησιάζοντας σιγά σιγά προς το τέλος της παρουσίασης, είμαστε πλέον σε θέση να απαντήσουμε και να συζητήσουμε μερικά ερωτήματα που τίθενται στην εκφώνηση.

Κατά την κατασκευή του προγράμματος, χρειάστηκε πολλές φορές να τρέξω τον κώδικα για να διαπιστώσω συγκεκριμένες λειτουργίες. Αυτό μου έδωσε την ευκαιρία να

έρθω σε επαφή με αρκετές παραλλαγές παραφράσεων των κειμένων και να συμπεράνω ότι οι ενσωματώσεις των λέξεων, κατά μέσο όρο, αποτυπώνουν αρκετά καλά το νόημα των δυο κειμένων μάλιστα σε ορισμένες περιπτώσεις ξεπλάγην από την ποιότητα του λόγου.

Όπως σε κάθε άλλη κατασκευή οποιουδήποτε προγράμματος, αντιμετωπίστηκαν αρκετές δυσκολίες κατά την υλοποίηση και του συγκεκριμένου. Μία από τις μεγαλύτερες προκλήσεις που αντιμετώπισα ήταν η εύρεση του κατάλληλου μοντέλου παράφρασης, δεδομένου ότι μιλάμε ταυτόχρονα για δύο κείμενα τα οποία δεν είναι ξεκάθαρα ως προς το νόημά τους. Ο παράγοντας αυτός περιέπλεξε την διαδικασία, καθώς έπρεπε να αναζητήσω μοντέλα τα οποία είναι καταλληλότερα για τέτοιου είδους κείμενα.

Η σημασιολογική ανακατασκευή κειμένων μπορεί να αυτοματοποιηθεί με τη χρήση μοντέλων NLP, όπως τα Pegasus, BART και T5(ramsrigouthamg), τα οποία δημιουργούν παραφράσεις διατηρώντας το νόημα του αρχικού κειμένου. Η αξιολόγηση της ομοιότητας γίνεται με embeddings και cosine similarity, ενώ η οπτικοποίηση των λέξεων μέσω PCA βοηθά στην κατανόηση των σημασιολογικών σχέσεων. Έτσι, η διαδικασία γίνεται αποδοτική, αντικειμενική και εύκολα επαναλήψιμη.

Κατά την εκτέλεση και την προβολή των αποτελεσμάτων παρατηρήσαμε , όπως φαίνεται και στην εικόνα 4, ότι το cosine similarity για όλες τις παραφράσεις των δύο κειμένων είναι υψηλό. Παρότι η βαθμολογίες είναι κοντά η ποιότητα του λόγου σε συνδυασμό με την μεταφορά του ίδιου νοήματος ξεπερνάει τις προσδοκίες μου, ειδικά στο πρώτο και στο τελευταίο μοντέλο παράφρασης. Μπορούμε λοιπόν να πούμε ότι υπάρχει μία ξεκάθαρη υπεροχή στην ποιότητα ανακατασκευής ανάμεσα στα Pegasus , Ramsrigouthamg και BART – large.

Συμπεράσματα

Η παρούσα μελέτη ανέδειξε τη σημασία και τη δυναμική της σημασιολογικής ανακατασκευής κειμένων μέσα από σύγχρονα εργαλεία της Επεξεργασίας Φυσικής Γλώσσας (NLP). Μέσω της χρήσης διαφορετικών μοντέλων παραφράσεων επιχειρήθηκε η δημιουργία επαναδιατυπωμένων κειμένων, ενώ παράλληλα μελετήθηκε η απόσταση μεταξύ των αρχικών και παραγόμενων προτάσεων, τόσο ποσοτικά μέσω του cosine similarity όσο και οπτικά με τη μέθοδο PCA. Ένα βασικό εύρημα ήταν ότι τα παραγόμενα κείμενα από τα εξελιγμένα προ-εκπαιδευμένα μοντέλα διατήρησαν υψηλό βαθμό σημασιολογικής ομοιότητας με τα αρχικά, επιβεβαιώνοντας την ικανότητα των νευρωνικών γλωσσικών μοντέλων να συλλαμβάνουν το νόημα πέρα από τη μορφή, παρότι σε μερικά σημεία υπήρχαν αστοχίες. Παράλληλα, η χρήση word embeddings σε συνδυασμό με τεχνικές οπτικοποίησης φανέρωσε διακριτές αλλά συγγενικές χωρικές κατανομές μεταξύ των λέξεων των αρχικών και παραφρασμένων προτάσεων. Ωστόσο, η διαδικασία παρουσίασε και προκλήσεις: η επιλογή κατάλληλων μοντέλων, η ανάγκη για καθαρισμό του κειμένου και η ερμηνεία των οπτικοποιημένων δεδομένων απαιτούν πολύωρη έρευνα και

τεχνική εξοικείωση. Επιπλέον, ορισμένες παραφράσεις παρότι εντός νοήματος (γιαυτό άλλωστε πετυχαίνουν ψηλό βαθμό), παρατηρείται να αλλοιώνουν πτυχές του αρχικού ύφους ή να προδίδουν τη συνοχή του λόγου.

Συνολικά όμως μπορούμε να συμπεράνουμε, ότι η εργασία αυτή ανέδειξε τις δυνατότητες αλλά και τις απαιτήσεις της εφαρμογής NLP για νοηματική παραλλαγή κειμένων και μας έκανε να φανταστούμε την ανάγκη και την εφαρμογή των μεθόδων της σε πιο σύνθετα προγραμματιστικά περιβάλλοντα.

Βιβλιογραφία

- 1) Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., ... & Zettlemoyer, L. (2020). *BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension*. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics* (pp. 7871–7880). <https://doi.org/10.18653/v1/2020.acl-main.703>
- 2) Ramsrigouthamg. (n.d.). *T5 paraphraser model*. Hugging Face. https://huggingface.co/ramsrigouthamg/t5_paraphraser
- 3) Reimers, N., & Gurevych, I. (2019). *Sentence-BERT: Sentence embeddings using Siamese BERT-networks*. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing* (pp. 3982–3992). <https://doi.org/10.18653/v1/D19-1410>
- 4) Singhal, A. (2001). *Modern information retrieval: A brief overview*. *IEEE Data Engineering Bulletin*, 24(4), 35–43.
- 5) Zhang, J., Zhao, Y., Saleh, M., & Liu, P. J. (2020). *PEGASUS: Pre-training with extracted gap-sentences for abstractive summarization*. arXiv preprint arXiv:1912.08777. <https://arxiv.org/abs/1912.08777>

