Επεξεργασία Φυσικής Γλώσσας

Απαλλακτική εργασία – Δομημένη Αναφορά

Σεπτέμβριος 2025

Ομάδα:

Αθανάσιος – Σταύρος Γούσιας Π22036

Γεώργιος Μαριέττος Π22096

GitHub: https://github.com/georgeMariettos/NLP_Project_2025

Περιεχόμενα

Εισαγωγή

Η σημασιολογική ανακατασκευή είναι μια τεχνική η οποία ανήκει στο πεδίο της σημασιολογικής ανάλυσης. Η σημασιολογική ανάλυση είναι ένας από τους πυλώνες πάνω στους οποίους στηρίζεται η Επεξεργασία της Φυσικής Γλώσσας(Natural Language Processing). Στόχος της σημασιολογικής ανάλυσης είναι η ανακατασκευή κειμένων ώστε το αποτέλεσμα να είναι συνεχές ως προς το νόημα και γραμματικά σωστό.

Στόχος της εργασίας είναι η σημασιολογική ανακατασκευή δύο κειμένων με συντακτικά λάθη και η ανάλυση της επίδοσης των διάφορων τεχνικών που θα χρησιμοποιηθούν για την ανακατασκευή αυτή. Για την ανακατασκευή θα χρησιμοποιηθούν αυτόματα καθώς και τρία διαφορετικά Μεγάλα Γλωσσικά Μοντέλα (Large Language Models).

Μεθοδολογία

Παραδοτέο 1: Ανακατασκευή των κειμένων

Ερώτημα Α

Για το πρώτο ερώτημα υλοποιήθηκε ένα πρόγραμμα που χρησιμοποιεί “regular expressions” για τον μετασχηματισμό των δύο προτάσεων. Για αυτήν την διαδικασία έχουμε ορίσει ένα σύνολο από κανόνες που εντοπίζουν γραμματικά ή συντακτικά λάθη στις προτάσεις που επιλέξαμε και τα αντικαταστούν με πιο ορθές και σαφείς φράσεις. Οι κανόνες αυτοί έχουν επιλεχθεί για την διόρθωση των δύο προτάσεων που επιλέξαμε συγκεκριμένα και ο στόχος είναι να ανακατασκευάσουν και τις δύο βελτιώνοντας την σαφήνεια και παράλληλα διατηρώντας το αρχικό τους νόημα. Η διαδικασία της ανακατασκευής ολοκληρώνεται όταν έχουν μετασχηματιστεί και οι δύο προτάσεις και δεν έχουν μείνει άλλοι κανόνες για να εφαρμοστούν. Τα αποτελέσματα αποθηκεύονται στα αρχεία reconstructed1.txt, reconstructed2.txt οπού θα χρησιμοποιηθούν για το επόμενο ερώτημα.

```python
sentence = "Today is our dragon boat festival, in our Chinese culture, to celebrate it with all safe and great in our lives."
sentence2 = "Anyway, I believe the team, although bit delay and less communication at recent days, they really tried best for paper and cooperation."

rules = [
    [re.compile(r"with all safe and great in our lives"), "safely and happily in our lives"],
    [re.compile(r"Today is our dragon boat festival"), "Today is our Dragon Boat festival"],
    [re.compile(r"in our Chinese culture, to celebrate it"), "in our Chinese culture, we celebrate it"],
    [re.compile(r"bit delay"), "a bit of a delay"],
    [re.compile(r"they really tried best"), "they really tried their best"],
    [re.compile(r"at recent days"), "in recent days"],
    [re.compile(r"for paper and cooperation"), "for the paper and our cooperation"],
    [re.compile(r"at recent days"), "in recent days"],
    [re.compile(r"I believe the team"), "I believe in the team"]
]

def reconstruct_sentence(sentence):  2 usages

    reconstructed_sentence = sentence

    for i in range(len(rules)):

        reconstructed_sentence = apply_rule(i, reconstructed_sentence)

    return reconstructed_sentence

def apply_rule(i, original_sentence):  1 usage

    pattern = rules[i][0]

    replacement = rules[i][1]

    sentence = re.sub(pattern, replacement, original_sentence)

    if sentence != original_sentence:

        print(f"{pattern.pattern} -> {replacement}")

    return sentence
```

Ερώτημα Β

Για το δεύτερο ερώτημα αξιοποιήθηκαν γλωσσικά μοντέλα από τη βιβλιοθήκη “Hugging face transformers” με τη μορφή pipelines τύπου “text2text-generation”. Τα μοντέλα που χρησιμοποιήθηκαν συγκεκριμένα είναι:

1. stanford-oval/paraphraser-bart-large (BART)
2. tuner007/pegasus_paraphrase (PEGASUS)
3. humarin/chatgpt_paraphraser_on_T5_base (ChatGPT_T5)

Το πρόγραμμα αρχικά διασπά τα κείμενα σε προτάσεις με την βοήθεια της βιβλιοθήκης nltk, για την ανακατασκευή των προτάσεων χρησιμοποιήθηκε η παραφραστική λειτουργία του κάθε μοντέλου (paraphrase) ώστε να διατηρηθεί το αρχικό νόημα και παράλληλα να διορθωθούν τα συντακτικά λάθη. Κατά την παραγωγή των ανακατασκευασμένων προτάσεων ορίστηκαν συγκεκριμένες παράμετροι για τον έλεγχο της ποικιλίας και της ποιότητας των αποτελεσμάτων. Ρυθμιστικέ η παράμετρος beam search (num_beams=5, num_beam_groups=5) ώστε το μοντέλο να εξετάσει πολλαπλές πιθανές εκδοχές πριν επιλέξει την τελική. Εφαρμόστηκε επίσης diversity_penalty=0.2 για να ενθαρρυνθούν διαφορετικές εκδοχές και να αποφευχθούν επαναλήψεις. Το όριο λέξεων τέθηκε με την παράμετρο max_new_tokens=80, ώστε να διασφαλιστεί ότι οι παραφράσεις παραμένουν επαρκώς αναλυτικές. Τέλος, ορίστηκε num_return_sequences=1 ώστε να παραχθεί μία παραλλαγή ανά πρόταση για κάθε μοντέλο. Τα αποτελέσματα αποθηκεύονται στα αρχεία “text1_version1.txt”, “text2_version2.txt”, “text1_version2.txt” κ.ο.κ.

```python
sentences_text1 = nltk.sent_tokenize(text1)
sentences_text2 = nltk.sent_tokenize(text2)


models = [
    ("stanford-oval/paraphraser-bart-large", "version1"),
    ("tuner007/pegasus_paraphrase", "version2"),
    ("humarin/chatgpt_paraphraser_on_T5_base", "version3"),
]

def paraphrase(sentences, text_id):  2 usages
    for model_name, version in models:
        paraphraser = pipeline( task: "text2text-generation", model=model_name)
        paraphrased_sentences = []
        for s in sentences:
            out = paraphraser(
                *args: s,
                do_sample=False,
                num_beams=5,
                num_beam_groups=5,
                max_new_tokens=80,
                diversity_penalty=0.2,
                num_return_sequences=1
            )
            paraphrased_sentences.append(out[0]["generated_text"])


        filename = f"text{text_id}_{version}.txt"

        with open(filename, "w", encoding="utf-8") as f:

            f.write("\n".join(paraphrased_sentences))

        print(f"Saved: {filename}")
```

Ερώτημα C

Για την διευκόλυνση της σύγκρισης των αποτελεσμάτων από το πρώτο και δεύτερο παραδοτέο, συγκρίνονται οι 2 ανακατασκευασμένες προτάσεις με τις 2 αντίστοιχες προτάσεις που παρήγαγε το κάθε μοντέλο, παράλληλα εφαρμόζουμε τις ίδιες τεχνικές και στις αρχικές προτάσεις ώστε να διακρίνουμε αν υπήρξε βελτίωση σημασιολογικά μετά την ανακατασκευή τους. Αρχικά, χρησιμοποιείται η βιβλιοθήκη “spaCy” σε συνδυασμό με το “benepar” για τη δημιουργία συντακτικών δέντρων (constituency trees). Τα δέντρα αυτά δείχνουν με ποιον τρόπο ομαδοποιούνται οι λέξεις σε φράσεις, παρέχοντας έτσι μια οπτική αναπαράσταση της συντακτικής συνοχής κάθε εκδοχής.

Στην συνέχεια, για τον υπολογισμό σκορ της κάθε πρότασης αξιοποιείται το προεκπαιδευμένο μοντέλο “CoLA (textattack/roberta-base-CoLA)”. Το μοντέλο αυτό επιστρέφει μια πιθανότητα που αντιπροσωπεύει τον βαθμό στον οποίο μια πρόταση θεωρείται γραμματικά αποδεκτή στα αγγλικά. Με τον τρόπο αυτό συγκρίνουμε τα αποτελέσματα όχι μόνο σε επίπεδο σύνταξης αλλά και σε επίπεδο γλωσσικής ορθότητας.

```
model_name = "textattack/roberta-base-CoLA"
tokenizer = AutoTokenizer.from_pretrained(model_name)
model = AutoModelForSequenceClassification.from_pretrained(model_name)

nltk.download('punkt', quiet=True)
benepar.download('benepar_en3')

original_sentence1 = "Today is our dragon boat festival, in our Chinese culture, to celebrate it with all safe and great in our lives."
original_sentence2 = "Anyway, I believe the team, although bit delay and less communication at recent days, they really tried best for paper and cooperation."

reconstructed_sentence1 = ""
reconstructed_sentence2 = ""

bart_sentence1 = "Today is the Dragon Boat Festival, and in Chinese culture, we celebrate it with everything safe and good in our lives."
pegasus_sentence1 = "The dragon boat festival is a celebration in our Chinese culture and we should all be happy."
humarin_sentence1 = "Our Chinese culture features a dragon boat festival today, designed to celebrate with all that is good and safe in our lives."

bart_sentence2 = "Anyway, I trust that the team, even if a little late and less communicative in the last few days, has really tried its best in terms of paper and cooperation."
pegasus_sentence2 = "I believe the team tried their best for paper and cooperation despite the recent delay and less communication."
humarin_sentence2 = "Despite experiencing some delays and less communication than in recent days, the team did well in terms of paper-based and collaborative issues."
```

```
with open("reconstructed1.txt", encoding="utf-8") as f:
    for line in f:
        reconstructed_sentence1 += line

with open("reconstructed2.txt", encoding="utf-8") as f:
    for line in f:
        reconstructed_sentence2 += line


def get_parser():  1 usage
    nlp = spacy.load("en_core_web_sm")

    if "benepar" not in nlp.pipe_names:

        nlp.add_pipe( factory_name: "benepar", config={"model": "benepar_en3"})

    return nlp

parser = get_parser()


def print_constituency_tree(text, sentence_name):  1 usage

    print(f"Constituency tree for {sentence_name}")

    doc = parser(text)

    for i, sent in enumerate(doc.sents, start=1):

        tree = Tree.fromstring(sent._.parse_string)

        print(f"\nSentence {i}: {sent.text}")

        tree.pretty_print()
```

```python
def cola_score(sentence):  1 usage

    inputs = tokenizer(sentence, return_tensors="pt", truncation=True)

    with torch.no_grad():
        logits = model(**inputs).logits

    return torch.softmax(logits, dim=-1)[0, 1].item()

def analyze_sentence(text, sentence_name):  6 usages
    print_constituency_tree(text, sentence_name)
    score = cola_score(text)
    print(f"CoLA score for {sentence_name}: {score:.4f}")


analyze_sentence(original_sentence1, sentence_name: "Original Sentence 1")
analyze_sentence(reconstructed_sentence1, sentence_name: "Reconstructed Sentence 1")
analyze_sentence(original_sentence2, sentence_name: "Original Sentence 2")
analyze_sentence(reconstructed_sentence2, sentence_name: "Reconstructed Sentence 2")

model_sentences1 = {
    "BART Result for Sentence 1": bart_sentence1,
    "PEGASUS Result for Sentence 1": pegasus_sentence1,
    "HUMARIN Result for Sentence 1": humarin_sentence1,
}

model_sentence2 = {
    "BART  Result for Sentence 2": bart_sentence2,
    "PEGASUS  Result for Sentence 2": pegasus_sentence2,
    "HUMARIN  Result for Sentence 2": humarin_sentence2,
}

for label, sent in model_sentences1.items():
    analyze_sentence(sent, label)

for label, sent in model_sentence2.items():
    analyze_sentence(sent, label)
```

Παραδοτέο 2: Υπολογιστική Ανάλυση (Semantic_Analysis.py)

Για την σύγκριση μεταξύ των έξι ανακατασκευασμένων κειμένων και των πρωτοτύπων, χρησιμοποιήθηκε το μοντέλο Doc2Vec για ενσωματώσεις παραγράφων της βιβλιοθήκης gensim. Το μοντέλο Doc2Vec υλοποιεί τον αλγόριθμο που δημοσίευσαν οι Lee και Mikolov το 2014[1,2].

---

[1] Řehůřek, R. (2024, August 10). Gensim: Topic modelling for humans. Doc2Vec Model - gensim. https://radimrehurek.com/gensim/auto_examples/tutorials/run_doc2vec_lee.html#doc2vec-model

[2] Le, Q., Mikolov, T., & Google Inc. (2014). Distributed representations of sentences and documents. In Proceedings of the 31st International Conference on Machine Learning (Vol. 32). https://cs.stanford.edu/~quocle/paragraph_vector.pdf

Για την μάθηση του μοντέλου χρησιμοποιήθηκε το dataset plain-text-wikipedia-simpleenglish(https://www.kaggle.com/datasets/ffatty/plain-text-wikipedia-simpleenglish?resource=download). Το μοντέλο έχει ρυθμιστεί έτσι ώστε να υπολογίζει διανύσματα μεγέθους εκατό, χρησιμοποιώντας το implementation PV-DBOW. Μέσω της μεθόδου infer_vector του εκπαιδευμένου μοντέλου, μπορούμε να βρούμε το διάνυσμα που αντιστοιχεί σε ολόκληρο το κείμενο. Με αυτόν τον τρόπο θα μπορέσουμε να βρούμε την ομοιότητα συνημίτονου ώστε να γίνει δυνατή η σύγκριση μεταξύ των αποτελεσμάτων του παραδοτέου 1.

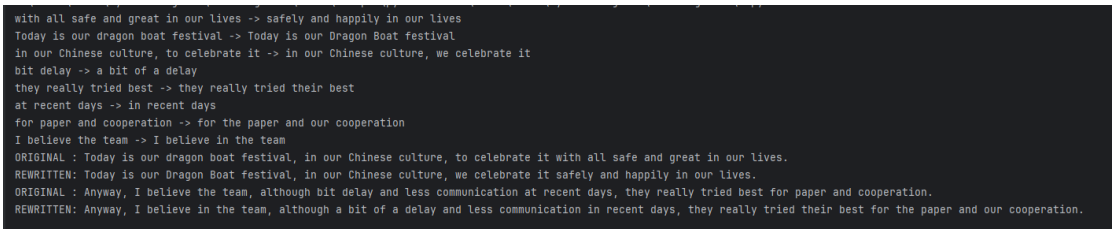
Στην συνέχεια, μπορούμε να απεικονίσουμε τις διαφορές μεταξύ των κειμένων μεταξύ τους, αλλά και με βάση τα διανύσματα λέξεων τους. Για την πραγματοποίηση της απεικόνισης αυτής χρησιμοποιείται Principal Component Analysis(PCA) για κάθε διάνυσμα που χρειάζεται να απεικονιστεί.

Αποτελέσματα

Παραδοτέο 1: Ανακατασκευή των κειμένων

Ερώτημα Α

Το πρόγραμμα αρχικά εκτυπώνει όλες τις αντικαταστάσεις στην κάθε πρόταση από τους κανόνες που εφαρμόστηκαν και στην συνέχεια τις αρχικές και ανακατασκευασμένες εκδόσεις των προτάσεων. Όπως ήταν αναμενόμενο οι στοχευμένες αλλαγές των κανόνων βελτίωσαν την σαφήνεια της κάθε πρότασης διορθώνοντας αποτελεσματικά τα συντακτικά λάθη με αποτέλεσμα να είναι πιο κατανοητές και ακριβείς.



Ερώτημα Β

Τα 3 μοντέλα παρήγαγαν αρκετά διαφορετικές εκδοχές των δύο κειμένων, όπως φαίνεται το BART διατήρησε σε μεγάλο βαθμό την δομή των αρχικών προτάσεων ενώ παράλληλα βελτίωσε την συνοχή. Το Pegasus έτεινε σε πιο απλουστευμένες εκδοχές με στόχο να βελτιώσει την σαφήνεια. Τέλος το ChatGPT_T5 παρήγαγε πιο δημιουργικές παραφράσεις βελτιώνοντας έτσι το ύφος με κόστος να απομακρυνθεί το νόημα από το αρχικό κείμενο σε ορισμένες περιπτώσεις.

Αποτελέσματα του μοντέλου Bart για τα δύο κείμενα (μπορούν επίσης να βρεθούν στα αρχεία που αναφέρθηκαν πριν):

Κείμενο 1:

«Today is the Dragon Boat Festival, and in Chinese culture, we celebrate it with everything safe and good in our lives.
I hope you enjoy this as much as I do, as is my deepest wish.
Thank you for the message that you showed our words to the doctor, as his next contractual check-up, for all of us.
I received this message so I could see the approved report.
Actually, I got a message from the Professor to show me this a few days ago.
I greatly appreciate the full support of the professor for our publication in Springer Proceedings»

Κείμενο 2:

“During our last discussion, I told him about the new submission we'd been waiting for since last fall, but the updates were confusing because they didn't include full feedback from the reviewers or maybe the editor?
Anyway, I trust that the team, even if a little late and less communicative in the last few days, has really tried its best in terms of paper and cooperation.
I think we should be grateful, I mean all of us, for accepting and trying until finally last week, the Springer connection.
Also, please remind me if the doctor still intends to edit the acknowledgements section before sending it again.
Since I haven't seen the final part yet, or perhaps I've missed it, I'm sorry if I did.
All in all, let's make sure everyone is safe and celebrate the outcome with strong coffee and future endeavors”

Αποτελέσματα του μοντέλου Pegasus για τα δύο κείμενα:

Κείμενο 1:

“The dragon boat festival is a celebration in our Chinese culture and we should all be happy.
Hope you enjoy it as I wish.
Thank you for your message, which will be shown to the doctor.
I received this message to see the approved one.
I received a message from the professor a couple of days ago, to show me.
The professor supported the Springer proceedings publication.”

Κείμενο 2:

“I told him about the new submission that we were waiting for but the updates were confusing as they did not include the full feedback from the reviewer or editor.

I believe the team tried their best for paper and cooperation despite the recent delay and less communication.
We should be grateful 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 that part final.
Let us make sure all are safe and celebrate the outcome with coffee and targets."

Αποτελέσματα του μοντέλου ChatGPT_T5 για τα δύο κείμενα:

Κείμενο 1:

"Our Chinese culture features a dragon boat festival today, designed to celebrate with all that is good and safe in our lives.
I hope you enjoy it as much as I do, and may it be filled with all my love and affection.
Thank you for your message to show us what we said to the doctor as his next contract checking.
I received this message to view the authorized message.
The professor sent me a message to show me this, which happened to be a few days ago.
The professor's complete backing for our Springer proceedings publication is greatly appreciated, especially since I am grateful for their complete support."

Κείμενο 2:

"During our last discussion, I shared with him the new submission I had been waiting for last autumn, but the changes were unclear as they did not provide full feedback from the reviewer or editor.
Despite experiencing some delays and less communication than in recent days, the team did well in terms of paper-based and collaborative issues.
I think we should all be grateful for the acceptance and efforts of all of us until the Springer link finally arrived last week.
In case the doctor decides to revise the acknowledgments section before resending, please keep this in mind.
Maybe I didn't catch the final part or something. I apologize, but I didn't.
Let us ensure the safety of all and celebrate the outcome with strong coffee and future targets."

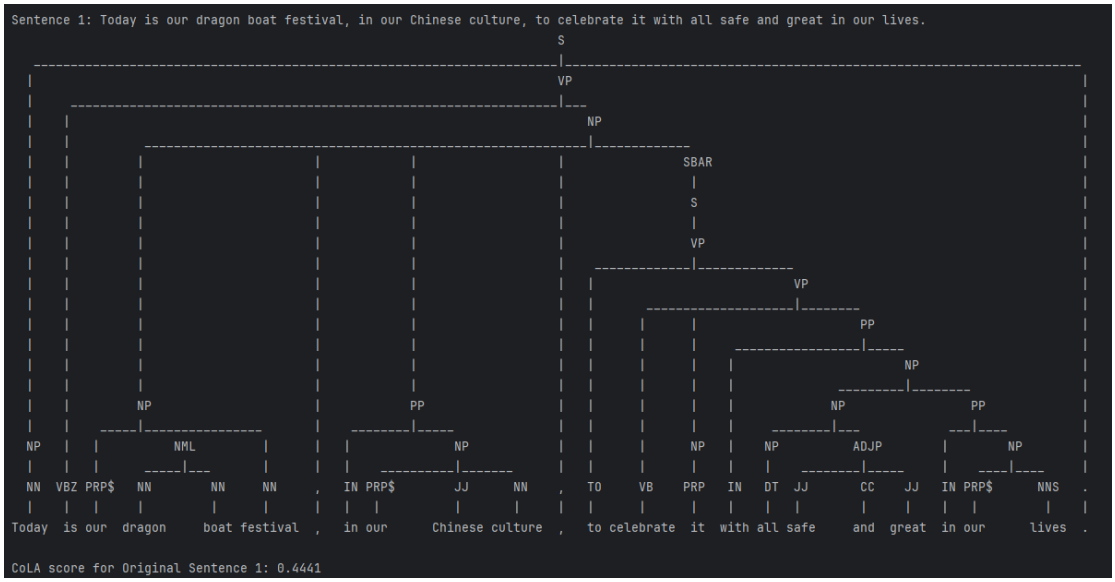
Ερώτημα C

Από τη σύγκριση των συντακτικών δέντρων και των CoLA scores παρατηρούμε ότι οι ανακατασκευασμένες προτάσεις είναι σαφώς βελτιωμένες συντακτικά και γραμματικά σε σχέση με τις αρχικές. Για παράδειγμα, η αρχική

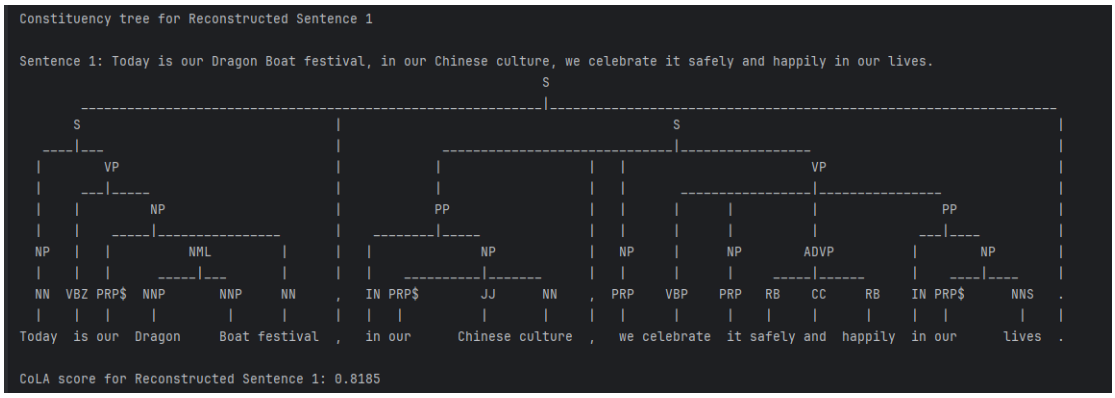
πρόταση 1 είχε χαμηλή βαθμολογία CoLA (0.44), ενώ η ανακατασκευασμένη πρόταση από το αυτόματο ανέβηκε σε 0.81, γεγονός που δείχνει βελτίωση στη γραμματική ορθότητα. Παρόμοια εικόνα παρατηρείται και στη δεύτερη πρόταση, όπου από 0.12 στην αρχική εκδοχή η ανακατασκευασμένη από το αυτόματο έφτασε στο 0.49.

Όσον αφορά τα τρία μοντέλα, το Pegasus εμφάνισε τις υψηλότερες βαθμολογίες (π.χ. 0.95 για το πρώτο κείμενο). Το BART πέτυχε επίσης υψηλά scores (0.90 και 0.70). Αντίθετα, το Humarin (T5) λόγω της μεγαλύτερης δημιουργικότητας στη διατύπωση με μικρότερη συνοχή, είχε τις βαθμολογίες (0.52 και 0.80).

Συντακτικό δέντρο + CoLA score για την αρχική πρόταση 1:

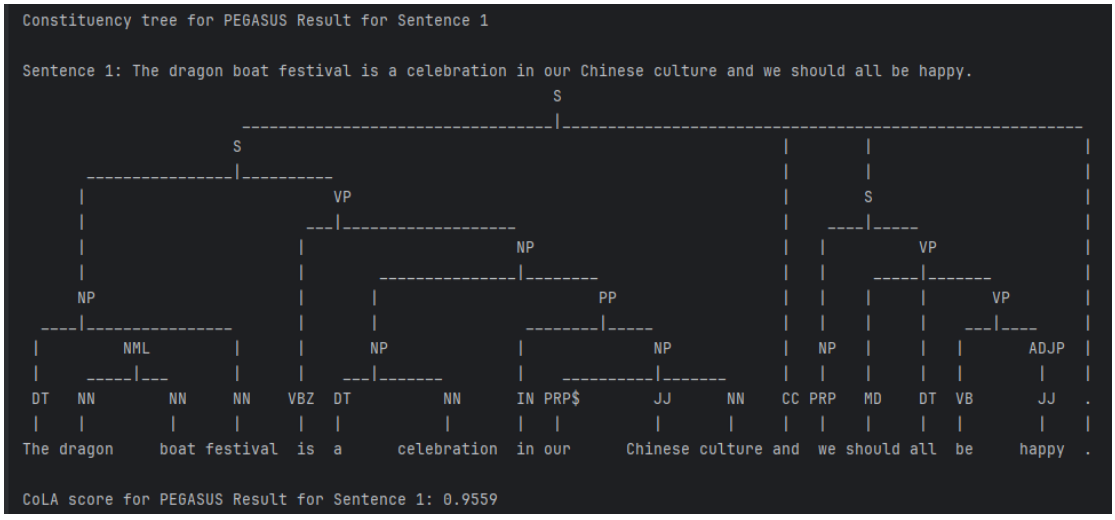


Συντακτικό δέντρο + CoLA score για την ανακατασκευασμένη (από το αυτόματο) πρόταση 1:

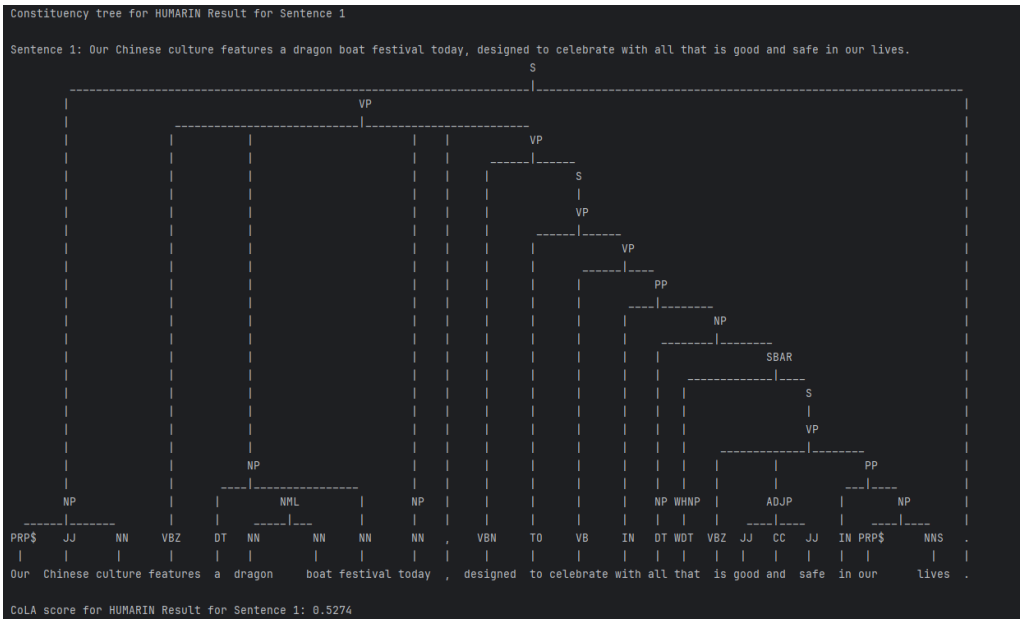


Συντακτικό δέντρο + CoLA score για την ανακατασκευασμένη (από το BART) πρόταση 1:

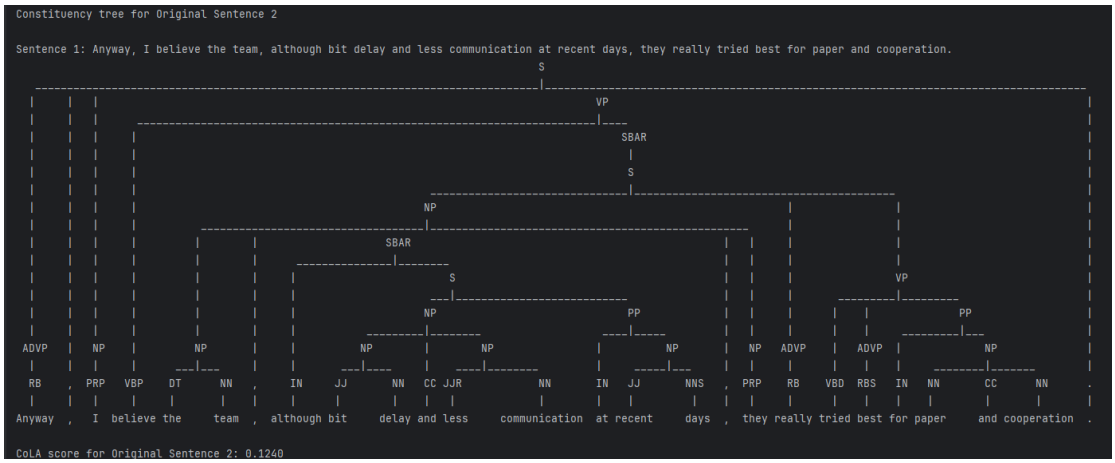
Συντακτικό δέντρο + CoLA score για την ανακατασκευασμένη (από το Pegasus) πρόταση 1:



Συντακτικό δέντρο + CoLA score για την ανακατασκευασμένη (από το ChatGPT_T5) πρόταση 1:



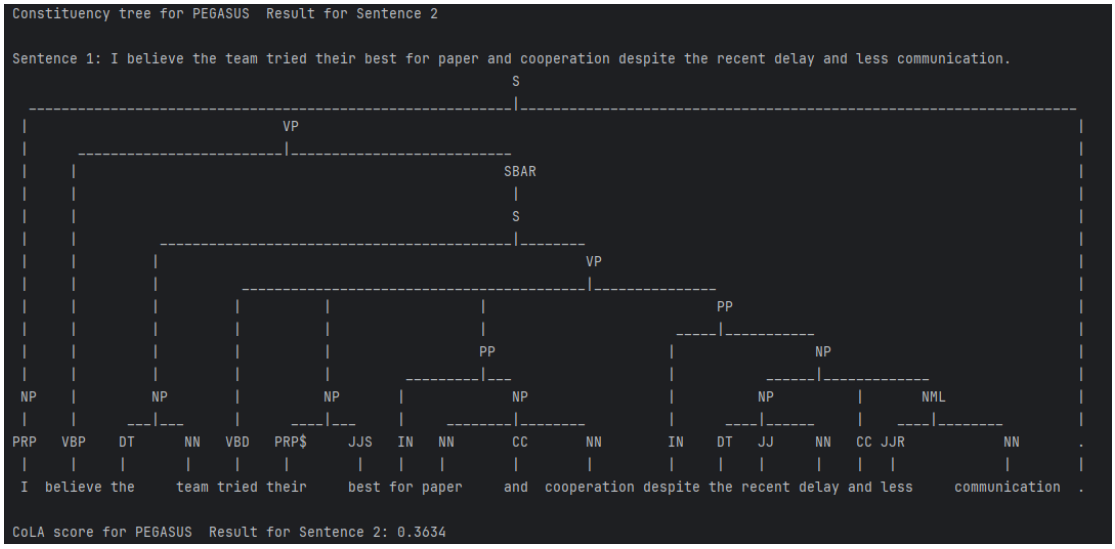
Συντακτικό δέντρο + CoLA score για την αρχική πρόταση 2:



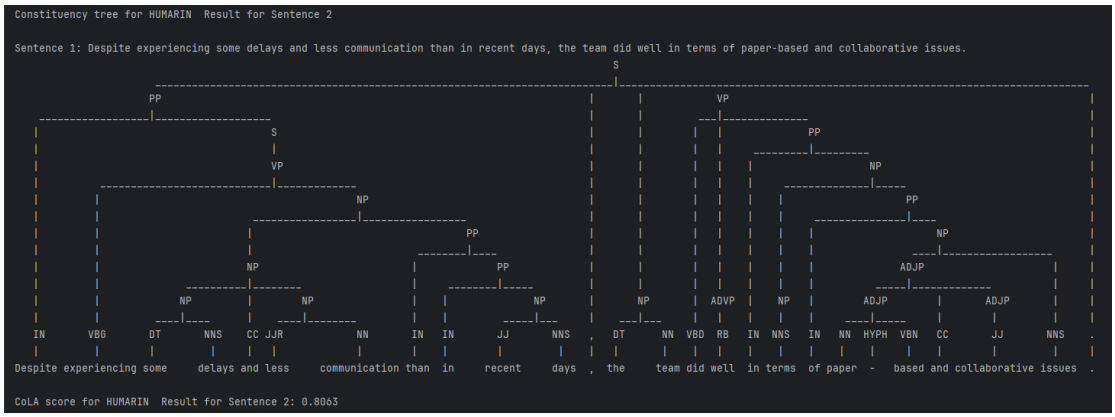
Συντακτικό δέντρο + CoLA score για την ανακατασκευασμένη (από το αυτόματο) πρόταση 2:

Συντακτικό δέντρο + CoLA score για την ανακατασκευασμένη (από το BART) πρόταση 2:

Συντακτικό δέντρο + CoLA score για την ανακατασκευασμένη (από το Pegasus) πρόταση 2:



Συντακτικό δέντρο + CoLA score για την ανακατασκευασμένη (από το ChatGPT_T5) πρόταση 2:



Παραδοτέο 2: Υπολογιστική Ανάλυση

Κείμενο 1:

Ομοιότητες συνημίτονου:

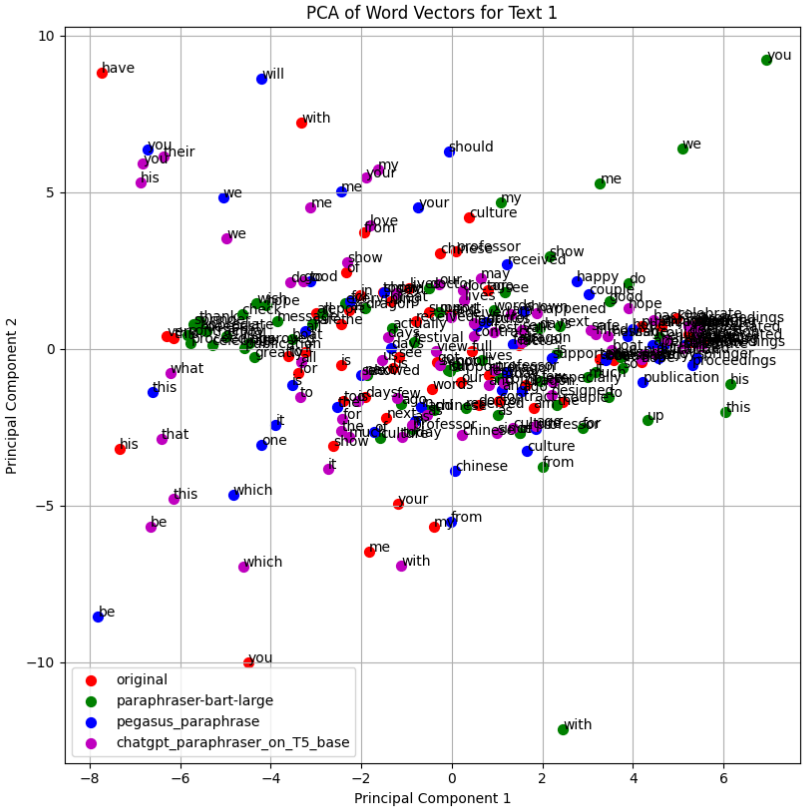
Τα κείμενα που παράχθηκαν στο παραδοτέο 1 συγκρίθηκαν με το πρωτότυπο και έδωσαν τις παρακάτω ομοιότητες συνημίτονου:

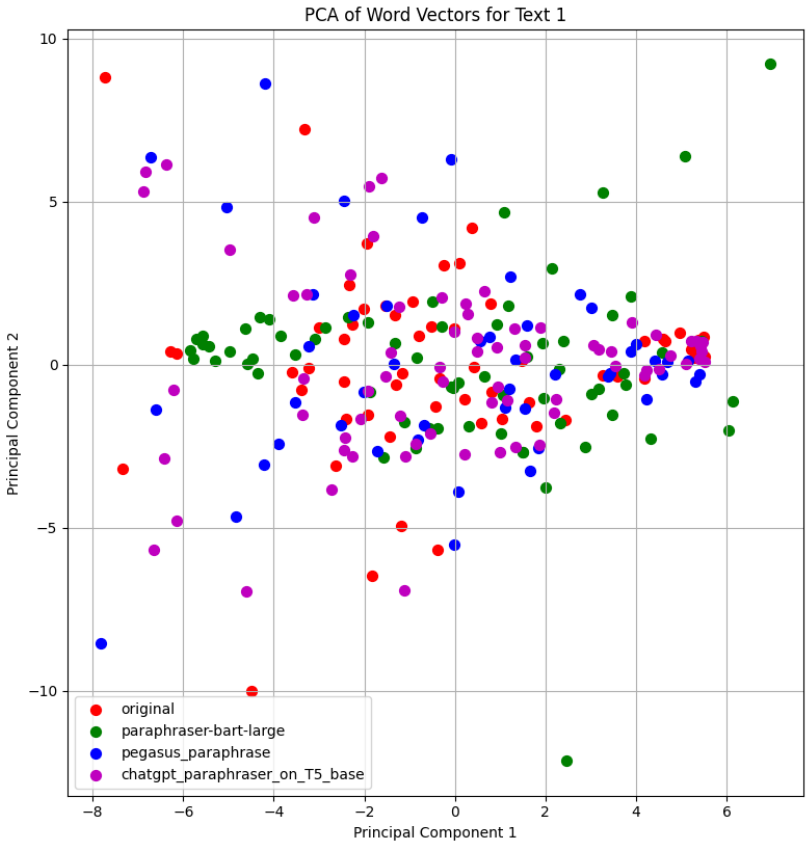
```
======Cosine similarity for Text 1======
Cosine similarity between (original, paraphraser-bart-large): 0.8962319513110509
Cosine similarity between (original, pegasus_paraphrase): 0.753082933796008
Cosine similarity between (original, chatgpt_paraphraser_on_T5_base): 0.8549924421300013
```

Από τα αποτελέσματα αυτά γίνεται αντιληπτό ότι το κείμενο που δημιούργησε το μοντέλο paraphraser-bart-large έχει μεγαλύτερη ομοιότητα με το πρωτότυπο, απ’ ότι τα υπόλοιπα κείμενα.

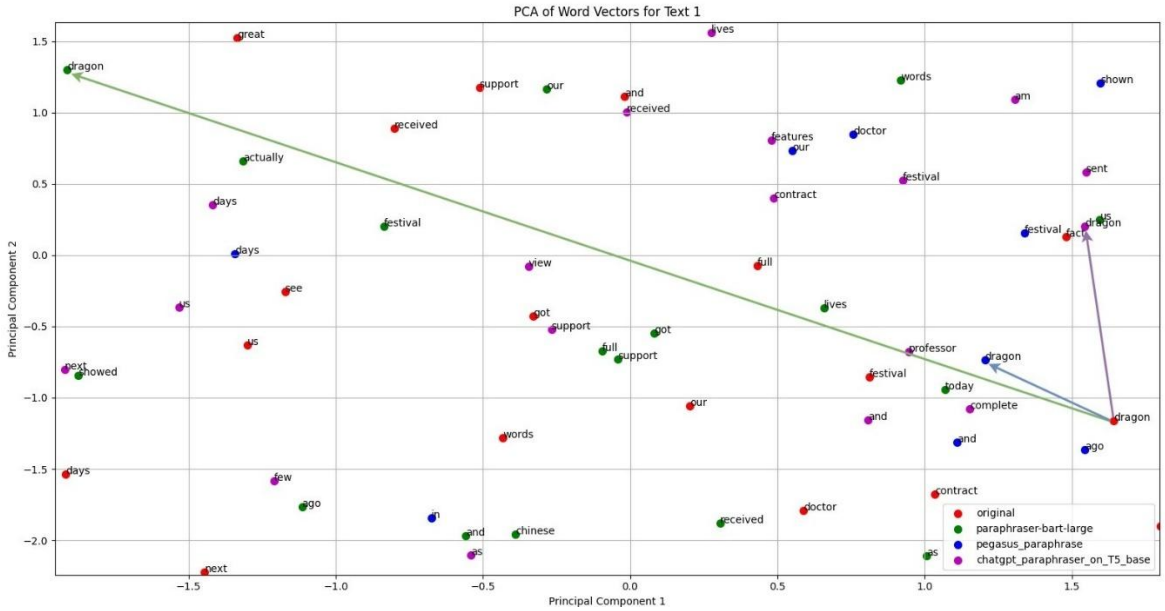
Αναπαράσταση των διανυσμάτων λέξεων στον σημασιολογικό χώρο:

Παίρνοντας για κάθε κείμενο τα διανύσματα λέξεων του δημιουργήθηκε το ακόλουθο γράφημα.



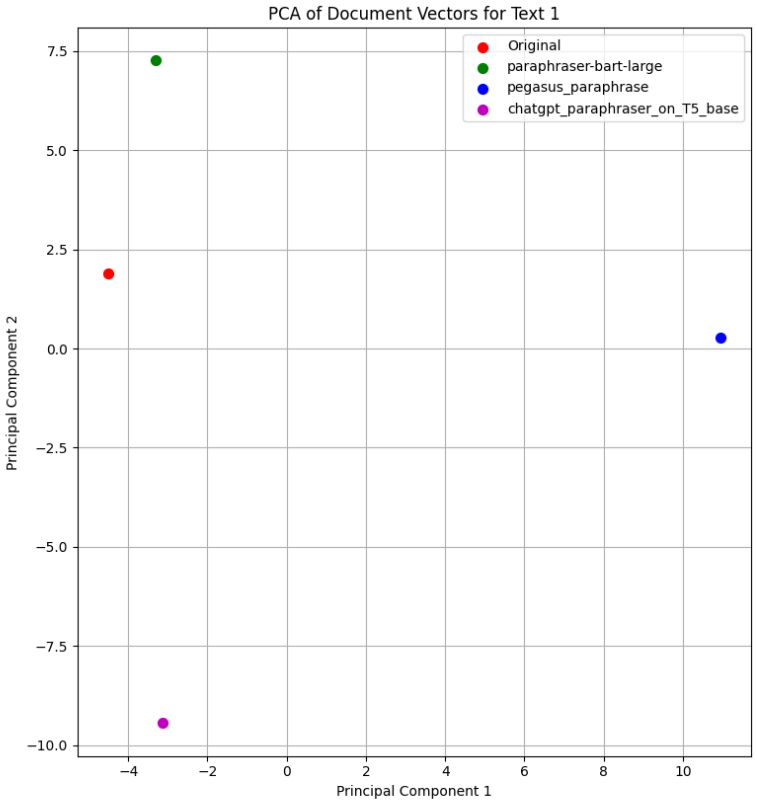


Αν επικεντρωθούμε σε συγκεκριμένες λέξεις μπορεί να γίνει αντιληπτή η σημασιολογική μετατόπιση που συνέβη κατά την ανακατασκευή. Για παράδειγμα, για τη λέξη «dragon» μπορεί να παρατηρηθεί μεγάλη μετατόπιση για το μοντέλο paraphraser-bart-large, ενώ τα υπόλοιπα μοντέλα είναι πιο «κοντά» στο αρχικό.



Αναπαράσταση των Document Vectors

Όπως φαίνεται στο παρακάτω σχήμα συνολικά το κείμενο του μοντέλου paraphraser-bart-large είναι πιο «κοντά» στο αρχικό κείμενο.



Κείμενο2:

Ομοιότητες συνημίτονου:

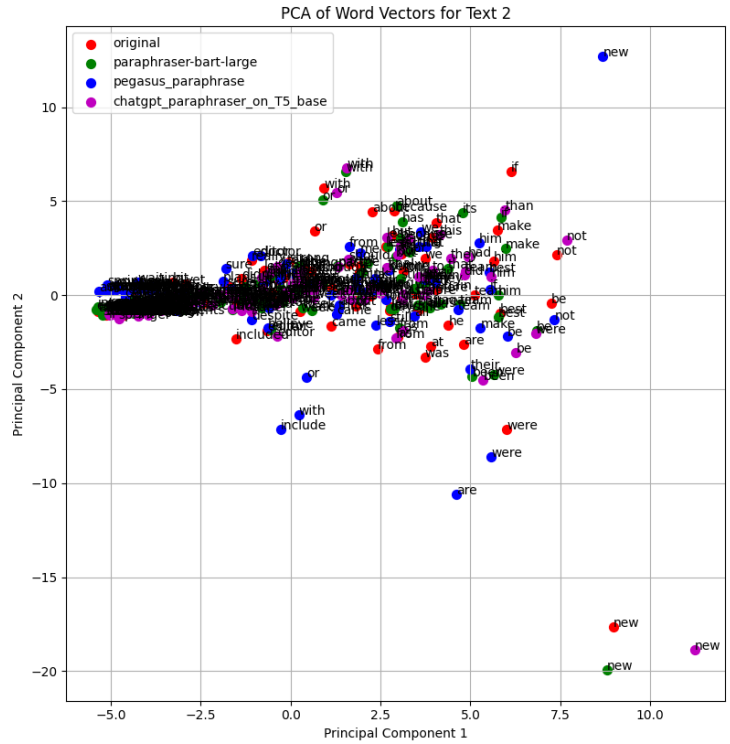
Τα κείμενα που παράχθηκαν στο παραδοτέο 1 συγκρίθηκαν με το πρωτότυπο και έδωσαν τις παρακάτω ομοιότητες συνημίτονου:

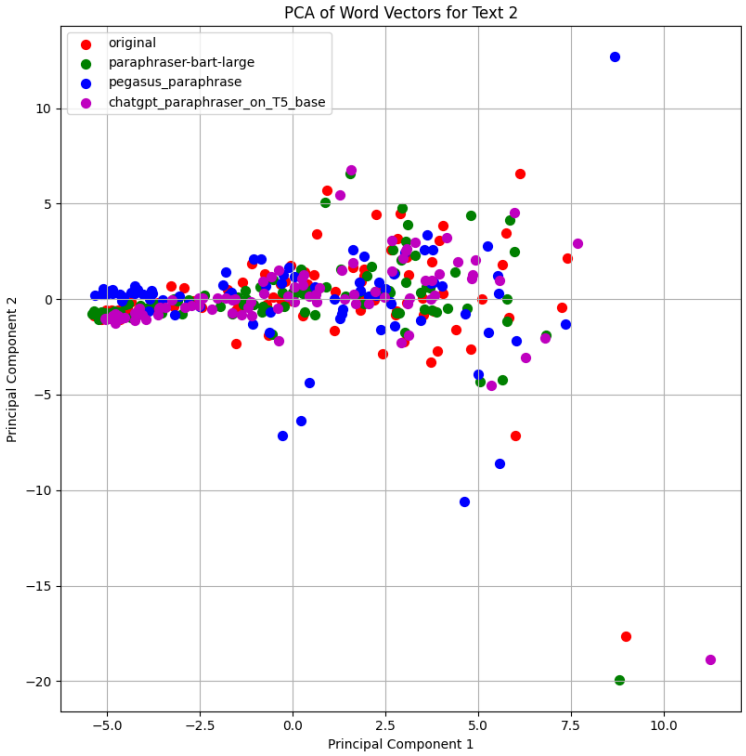
```
======Cosine similarity for Text 2======
Cosine similarity between (original, paraphraser-bart-large): 0.8807632108257559
Cosine similarity between (original, pegasus_paraphrase): 0.8840408220102391
Cosine similarity between (original, chatgpt_paraphraser_on_T5_base): 0.8808140982776788
```

Από τα αποτελέσματα αυτά γίνεται αντιληπτό ότι το κείμενο που δημιούργησε το μοντέλο Pegasus_paraphrase έχει μεγαλύτερη ομοιότητα με το πρωτότυπο, απ' ότι τα υπόλοιπα κείμενα.

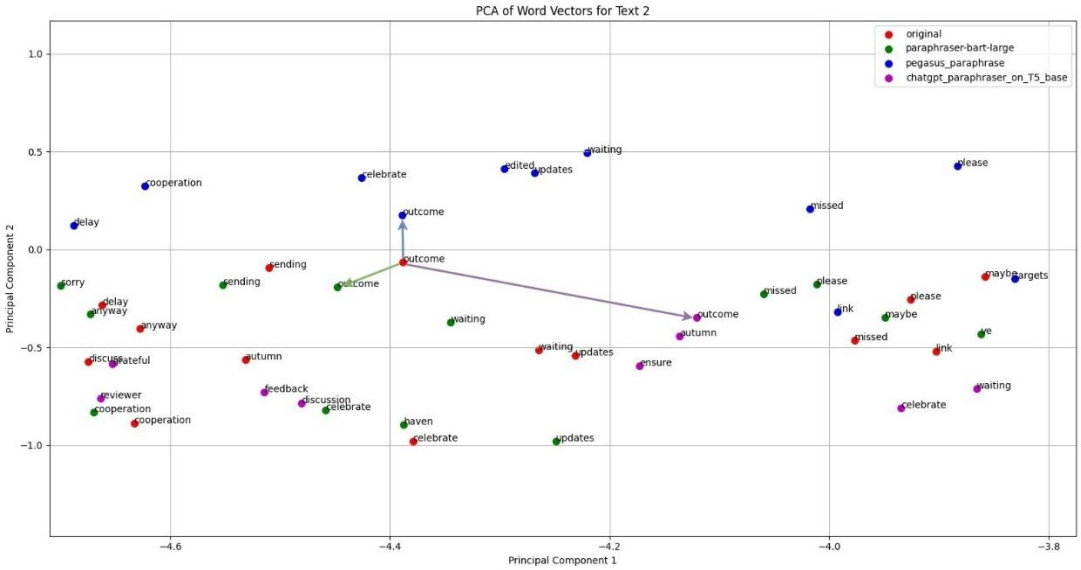
Αναπαράσταση των διανυσμάτων λέξεων στον σημασιολογικό χώρο:

Παίρνοντας για κάθε κείμενο τα διανύσματα λέξεων του δημιουργήθηκε το ακόλουθο γράφημα.


PCA of Word Vectors for Text 2
original
paraphraser-bart-large
pegasus_paraphrase
chatgpt_paraphraser_on_T5_base
Principal Component 2
Principal Component 1

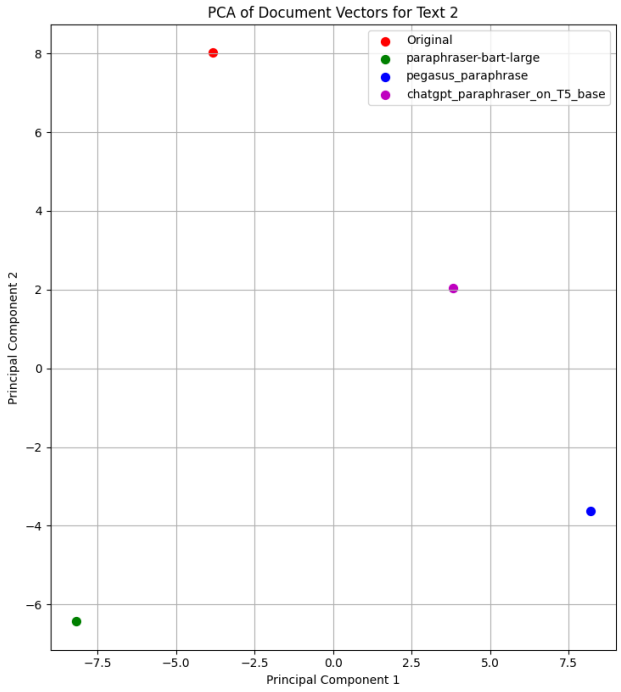

PCA of Word Vectors for Text 2
original
paraphraser-bart-large
pegasus_paraphrase
chatgpt_paraphraser_on_T5_base
Principal Component 2
Principal Component 1

Αν επικεντρωθούμε σε συγκεκριμένες λέξεις μπορεί να γίνει αντιληπτή η σημασιολογική μετατόπιση που συνέβη κατά την ανακατασκευή. Για παράδειγμα, για τη λέξη «outcome» μπορεί να παρατηρηθεί μεγάλη μετατόπιση για το μοντέλο chatgpt_paraphraser_on_T5_base, ενώ τα υπόλοιπα μοντέλα είναι πιο «κοντά» στο αρχικό.



Αναπαράσταση των Document Vectors

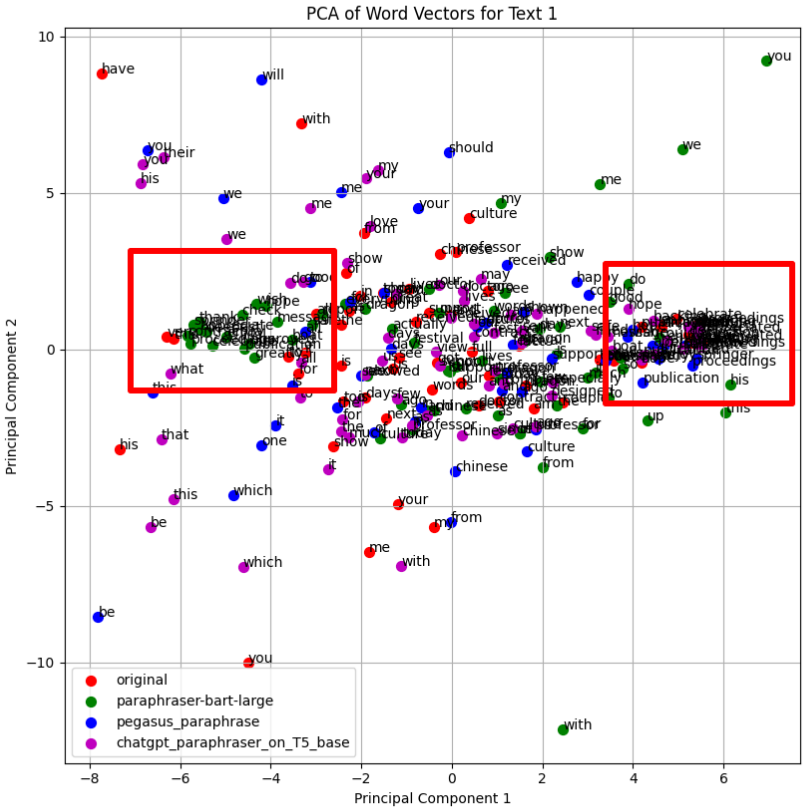
Όπως φαίνεται στο παρακάτω σχήμα συνολικά το κείμενο του μοντέλου paraphraser-bart-large είναι πιο «κοντά» στο αρχικό κείμενο.



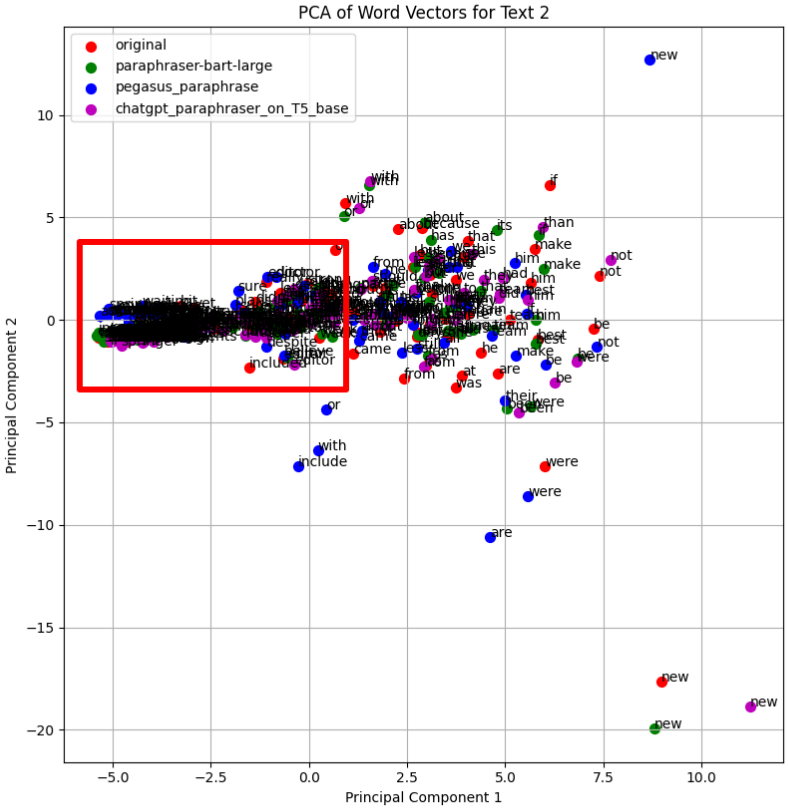
Ερμηνεία των αποτελεσμάτων

Διανύσματα λέξεων

Σχετικά με τα γραφήματα των διανυσμάτων των λέξεων, παρατηρείται μια βασική διαφορά μεταξύ τους. Για το κείμενο 1, Τα σημεία των λέξεων είναι σε μεγάλο βαθμό αραιά σκορπισμένα, εκτός από δύο μικρές περιοχές, όπως φαίνεται παρακάτω.



Στο κείμενο 2 όμως, φαίνεται ότι υπάρχει μεγάλη συνεκτικότητα σε μια περιοχή του γραφήματος.



Το γεγονός αυτό συμβαδίζει με τα αποτελέσματα της ομοιότητας των συνημίτονων.

Για το κείμενο 1, τα μοντέλα είχαν αισθητές διαφορές στην ομοιότητά τους σε σχέση με το πρωτότυπο. Για παράδειγμα, το πρώτο σε ομοιότητα απέχει από το τελευταίο κατά 0,143 περίπου.

Αντίθετα στο κείμενο 2 όλα τα κείμενα έχουν περίπου την ίδια ομοιότητα. Αυτό έχει ως αποτέλεσμα το πρώτο σε ομοιότητα να απέχει μόλις 0,0033.

Τα παραπάνω δείχνουν ότι ίσως για το πρώτο τα μοντέλα είχαν μεγαλύτερη δυσκολία στο να αναλύσουν και να ανακατασκευάσουν το κείμενο.

Βιβλιογραφία

1. Řehůřek, R. (2024, August 10). Gensim: Topic modelling for humans. Doc2Vec Model - gensim. https://radimrehurek.com/gensim/auto_examples/tutorials/run_doc2vec_lee.html#doc2vec-model
2. Le, Q., Mikolov, T., & Google Inc. (2014). Distributed representations of sentences and documents. In Proceedings of the 31st International

Conference on Machine Learning (Vol. 32). https://cs.stanford.edu/~quocle/paragraph_vector.pdf
3. GeeksforGeeks. (n.d.). Understanding Semantic Analysis NLP. https://www.geeksforgeeks.org/nlp/understanding-semantic-analysis-nlp/
4. Galarnyk, M. (2024, February 23). PCA using Python: A tutorial. Built In. https://builtin.com/machine-learning/pca-in-python
5. Holtz, Y. (n.d.). Visualise principal component analysis with Matplotlib. The Python Graph Gallery. https://python-graph-gallery.com/515-intro-pca-graph-python/
6. Wikipedia contributors. (2025, September 17). Cosine similarity. Wikipedia. https://en.wikipedia.org/wiki/Cosine_similarity
7. GeeksforGeeks. (2025, July 11). Principal Component Analysis(PCA). GeeksforGeeks. https://www.geeksforgeeks.org/data-analysis/principal-component-analysis-pca/