ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΤΗΛΕΠΙΚΟΙΝΩΝΙΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Επεξεργασία Φυσικής Γλώσσας

Απαλλακτική εργασία Εμβόλιμης εξεταστικής Δεκεμβρίου 2025

Παραδοτέο 3 – Δομημένη αναφορά

ΣΤΑΥΡΟΣ ΠΑΠΟΥΤΣΗΣ
Π21193

ΠΕΙΡΑΙΑΣ
06 Δεκεμβρίου 2025

Περιεχόμενα

Εισαγωγή

Η παρούσα εργασία εστιάζει στη σημασιολογική ανακατασκευή κειμένων με χρήση τεχνικών Επεξεργασίας Φυσικής Γλώσσας (Natural Language Processing – NLP). Στόχος της είναι η μετατροπή μη δομημένων ή σημασιολογικά αμφίβολων κειμένων σε σαφείς, ορθές και καλά δομημένες εκδοχές, διατηρώντας παράλληλα το αρχικό τους νόημα. Η σημασιολογική ανακατασκευή αποτελεί κρίσιμο πεδίο του NLP, καθώς εφαρμόζεται σε τομείς όπως η αυτόματη διόρθωση κειμένου με σκοπό τη βελτίωση της γραμματικής και της σύνταξης, η μηχανική μετάφραση για την παραγωγή ευανάγνωστων και φυσικών μεταφράσεων, τα συστήματα διαλόγου που απαιτούν κατανόηση και παραγωγή φυσικής γλώσσας, καθώς και η ανάλυση συναισθήματος, όπου επιδιώκεται η εξαγωγή νοήματος από ασαφή ή μη ξεκάθαρα διατυπωμένα κείμενα.

Για την υλοποίηση της σημασιολογικής ανακατασκευής, το NLP παρέχει ένα σύνολο απαραίτητων εργαλείων και τεχνικών. Η διαδικασία περιλαμβάνει τον διαχωρισμό του κειμένου σε βασικές μονάδες μέσω tokenization, την ανάθεση γραμματικών ετικετών με τη χρήση POS tagging, καθώς και την εύρεση της βασικής μορφής των λέξεων μέσω lemmatization. Επιπλέον, αξιοποιούνται τεχνικές αναπαράστασης λέξεων σε διανυσματικό χώρο μέσω word embeddings, ενώ σύγχρονα μοντέλα μετασχηματιστών (transformer models) επιτρέπουν την κατανόηση και παραγωγή κειμένου με τη χρήση μεθόδων βαθιάς μάθησης. Οι παραπάνω τεχνικές αξιοποιούνται επιλεκτικά, ανάλογα με το επίπεδο ανακατασκευής (πρόταση ή κείμενο).

Μεθοδολογία

Μεθοδολογία Παραδοτέου 1

Για το παραδοτέο 1 η αρχιτεκτονική της εργασίας βασίζεται σε ξεχωριστά τμήματα που συνθέτουν την ολοκληρωμένη μορφή του προγράμματος. Η αρχιτεκτονική αυτή επιτρέπει την επέκταση των ξεχωριστών τμημάτων και κάνει το debug πιο διαχειρίσιμο. Η λογική του αυτομάτου για το 1Α βασίζεται σε 3 στάδια υλοποίησης: την προ-επεξεργασία, την συντακτική ανάλυση και την γραμματική ανάλυση (η οποία περιλαμβάνει και μια μορφοποίηση post-processing). Συνολικά υλοποιείται μια αυτόματη ανακατασκευή των 2 επιλεγμένων προτάσεων βάσει κανόνων και επιφανειακές τεχνικές NLP. Η προσέγγιση εστιάζει περισσότερο στην γραμματική μορφή και βασική συντακτική οργάνωση της πρότασης. Δεν χρησιμοποιήθηκαν προεκπαιδευμένα μοντέλα ή σημασιολογικές αναπαραστάσεις. Για το παραδοτέο 1Β ζητείται στην εκφώνηση η δημιουργία 3 διαφορετικών pipeline για την επεξεργασία των 2 κειμένων.

Ακολουθήθηκε παρόμοια αρχιτεκτονική όπως το 1Α ως προς την επικοινωνία με την main συνάρτηση αλλά εδώ χρησιμοποιούνται προ-εκπαιδευμένα μοντέλα. Οι τρεις προσεγγίσεις επιλέχθηκαν ώστε να καλύψουν διαφορετικά επίπεδα NLP επεξεργασίας: επιφανειακή γλωσσική ανάλυση (TextBlob), σημασιολογική αναπαράσταση μέσω διανυσμάτων (embeddings) και βαθιά συμφραζόμενη ανακατασκευή μέσω transformer. Με τον τρόπο αυτό, το ίδιο πρόβλημα αντιμετωπίζεται με διαφορετικές υπολογιστικές φιλοσοφίες, επιτρέποντας ουσιαστική σύγκριση των μεθόδων.

Τα script επικοινωνούν με την κεντρική main συνάρτηση που συντονίζει τα επιμέρους στάδια. Η συνάρτηση main παίρνει από ξεχωριστά txt αρχεία το περιεχόμενο και το δίνει ως string σε μεταβλητές sentence 1,2 και text 1,2 για να προχωρήσει στην επεξεργασία. Διαθέτει ένα display menu ώστε να παίρνει εισαγωγή από το χρήστη αν πρέπει να τρέξει το Α, Β ή και τα δύο αφού ελέγξει πρώτα ότι τα directories υπάρχουν με την ensure_directories. Οι συναρτήσεις που επικοινωνούν με τα υπόλοιπα scripts ονομάζονται run_sentence_pipeline και run_text_pipeline και λειτουργούν με αντίστοιχο τρόπο. Δίνουν σε dictionary για sentences ή text το κείμενο και στη συνέχεια προχωρούν σε επαναλήψεις για την ανακατασκευή με βάση τα στοιχεία του dictionary. Όταν ολοκληρωθεί η διαδικασία εμφανίζουν τα αποτελέσματα και τα αποθηκεύουν στους αντίστοιχους data/results φακέλους για κάθε ερώτημα.

Τα directories του προγράμματος είναι τα παρακάτω:

Μεταβλητή	Path	Περιεχόμενο
BASE_DIR	data	φάκελος root data
RAW_DIR	data/raw	μη-επεξεργασμένα αρχεία
SENTENCES_DIR	data/raw/sentences	αρχεία με τις 2 προτάσεις
TEXTS_DIR	data/raw/texts	αρχεία με τα 2 κείμενα
RESULTS_DIR	data/results	όλα τα αποτελέσματα
SENTENCE_RESULTS_DIR	data/results/sentence_pipeline	αποτελέσματα 1Α
TEXTS_RESULTS_DIR	data/results/text_pipelines	αποτελέσματα 1Β

Παραδοτέο 1Α

Preprocessing.py

Για να μπορέσουμε να περάσουμε στην ανακατασκευή πρέπει πρώτα οι προτάσεις να βρίσκονται στην κατάλληλη μορφή. Το αρχικό κείμενο είναι μη δομημένο και θορυβώδες και στην κανονικοποίηση αυτή συμβάλει το στάδιο του preprocessing. Το πρόγραμμα αφαιρεί τις συντομεύσεις από το κείμενο (π.χ. Don’t -> Do not), μετατρέπει χαρακτήρες σε πεζά, αφαιρεί σημεία στίξης και διπλότυπα κενά. Το lowercasing των χαρακτήρων εξαλείφει τεχνητές διαφορές μεταξύ των λέξεων που διαφέρουν μόνο ως προς ένα γράμμα. Η αφαίρεση των σημείων στίξης και ειδικών χαρακτήρων μειώνει το θόρυβο καθώς οι χαρακτήρες αυτοί δεν συμβάλλουν στην σημασιολογική ανάλυση των λέξεων. Κατά την διαδικασία της αφαίρεσης δημιουργούνται διπλότυπα κενά οπότε αφαιρούνται και αυτά για να έχουμε ένα ομοιόμορφο αποτέλεσμα. Στη συνέχεια έχουμε tokenization με τη χρήση του NLTK. Έπειτα για κάθε λέξη βρίσκουμε την ετικέτα POS (Part-Of-Speech) ώστε να αποδοθούν γραμματικές ετικέτες σε κάθε token, παρέχοντας βασική συντακτική πληροφορία που είναι απαραίτητη για μεταγενέστερα στάδια ανάλυσης. Τέλος εφαρμόζεται lemmatization (lemma) – λημματοποίηση με σκοπό την μετατροπή των λέξεων στη βασική τους μορφή. Η διαδικασία γίνεται μετά το tokenization και POS tagging ώστε να διατηρείται το νόημα ενώ μειώνεται η μορφολογική ποικιλία. Με αυτό το τρόπο έχουμε δημιουργήσει μια λίστα με ανάλυση της κάθε λέξης για να την επεξεργαστούμε στη συνέχεια.

Syntactical_analysis.py

Αφού ολοκληρωθεί το στάδιο του preprocessing και παραχθεί το νέο string, στο επόμενο στάδιο απαιτείται η ανασύνταξή του με βάση τα tokens. Η συντακτική ανάλυση υλοποιείται σε επιφανειακό επίπεδο, χωρίς την εφαρμογή πλήρους συντακτικού parsing, και βασίζεται αποκλειστικά σε μορφοσυντακτική πληροφορία. Θεμέλιο της διαδικασίας αποτελούν τα POS tags που έχουν ήδη παραχθεί, καθώς λειτουργούν ως βασικό εργαλείο για την κατανόηση και αναδόμηση της δομής μιας πρότασης. Η αναγνώριση ονοματικών φράσεων πραγματοποιείται μέσω απλών POS patterns, τα οποία αντιστοιχούν στη διαδικασία του chunking, μια τυπική μορφή shallow parsing όπως παρουσιάζεται στο βιβλίο [1]. Αντίστοιχα, οι ρηματικές ομάδες εντοπίζονται με βάση τις γραμματικές ετικέτες και διαχωρίζονται σε κύρια και βοηθητικά ρήματα, πρακτική που ευθυγραμμίζεται με την επιφανειακή POS ανάλυση των ρηματικών δομών.

Η εξαγωγή της δομής SVO (Subject–Verb–Object) χρησιμοποιείται ως λειτουργική αναπαράσταση της πρότασης και αποτελεί βασικό σχήμα δομικής και πληροφοριακής ανάλυσης χωρίς την ανάγκη πλήρους γραμματικής περιγραφής. Οι προσθετικές φράσεις αντιμετωπίζονται ως ανεξάρτητες συντακτικές μονάδες της μορφής IN + NP, σύμφωνα με τη διάκριση φραστικών δομών που υιοθετείται στο πλαίσιο της επιφανειακής συντακτικής ανάλυσης. Παράλληλα, ο εντοπισμός δευτερεύουσων προτάσεων βασίζεται στην αναγνώριση ρητών υποδεέστερων συνδετικών λέξεων, τεχνική που θεωρείται απλή αλλά αποδεκτή μορφή συντακτικής τμηματοποίησης πριν από το πλήρες parsing.

Η αναδιάταξη της πρότασης πραγματοποιείται μέσω κανόνων που βασίζονται αποκλειστικά στη συντακτική πληροφορία, όπως τα POS tags και οι αναγνωρισμένες φράσεις, χωρίς αξιοποίηση σημασιολογικών χαρακτηριστικών. Στο πλαίσιο αυτό, δεν χρησιμοποιούνται Context-Free Grammars, dependency trees ή semantic role labeling, καθώς τα συγκεκριμένα εργαλεία ανήκουν σε πιο σύνθετα στάδια συντακτικής και σημασιολογικής ανάλυσης. Η συνολική προσέγγιση είναι κανονοκεντρική (rule-based), ερμηνεύσιμη και αναπαράξιμη, γεγονός που αποτελεί βασικό πλεονέκτημα για ένα πρώιμο στάδιο ανάλυσης στο πεδίο της Επεξεργασίας Φυσικής Γλώσσας.

Grammatical_correction.py

Στο Στάδιο 3 της επεξεργασίας εφαρμόζεται η γραμματική κανονικοποίηση (Grammatical Correction), η οποία ακολουθεί τη συντακτική ανακατασκευή και λειτουργεί ως στάδιο post-processing με στόχο τη βελτίωση της τελικής εξόδου. Η διαδικασία υλοποιείται ως επιφανειακή κανονικοποίηση κειμένου (surface-level grammatical normalization) και όχι ως πλήρης γραμματική ή σημασιολογική ανάλυση, σύμφωνα με τη λογική που περιγράφεται στο βιβλίο [2]. Η διόρθωση βασίζεται αποκλειστικά σε κανόνες και όχι σε στατιστικά ή μαθησιακά μοντέλα, γεγονός που την καθιστά ερμηνεύσιμη και ελεγχόμενη.

Η γραμματική κανονικοποίηση στηρίζεται σε τρεις βασικούς γραμματικούς κανόνες. Ο πρώτος αφορά τη συμφωνία υποκειμένου–ρήματος, όπου ένα υποκείμενο σε ενικό αριθμό απαιτεί ρήμα σε ενικό, όπως στη διόρθωση της πρότασης “The dog run” σε “The dog runs”. Ο δεύτερος κανόνας σχετίζεται με τη μορφολογική συνέπεια, διασφαλίζοντας τη διατήρηση ενιαίου χρόνου μέσα στην ίδια πρόταση, όπως στη μετατροπή της φράσης “He walked and talks” σε “He walked and talked” (όχι σημασιολογική εξαγωγή αλλά μορφολογική κανονικοποίηση για λόγους συνέπειας). Ο τρίτος κανόνας αφορά τη συμφωνία άρθρου–ουσιαστικού, διορθώνοντας ασυμβατότητες μεταξύ αριθμού και προσδιοριστή, όπως στις περιπτώσεις “a apples” ή “this books”, οι οποίες αναδομούνται σε γραμματικά ορθές μορφές.

Οι κανόνες αυτοί αξιοποιούν τις ετικέτες POS για την αναγνώριση και τον καθαρισμό αδόκιμων ακολουθιών λέξεων, ενώ η υλοποίηση υποστηρίζεται από εργαλεία όπως το NLTK για POS tagging, το WordNetLemmatizer για μείωση μορφολογικών παραλλαγών και προσαρμοσμένα regex patterns για τον εντοπισμό επαναλήψεων και ανωμαλιών. Παράλληλα, εφαρμόζονται συμπληρωματικοί κανόνες για την αφαίρεση διπλότυπων λέξεων, πλεονασμών και «ορφανών» ουσιαστικών που δεν εντάσσονται σωστά στη δομή της πρότασης.

Τέλος, το στάδιο του post-processing περιλαμβάνει την τελική μορφοποίηση του κειμένου, όπως την αφαίρεση περιττών κενών, τη σωστή χρήση σημείων στίξης και τη μετατροπή του πρώτου γράμματος κάθε πρότασης σε κεφαλαίο, ώστε το παραγόμενο κείμενο να είναι γραμματικά ορθό, συνεπές και αναγνώσιμο.

Παραδοτέο 1Β

Pipeline 1: TextBlob Library

Το Pipeline 1 βασίζεται σε μια στατιστική προσέγγιση και υλοποιείται με τη χρήση της βιβλιοθήκης TextBlob, η οποία αναφέρεται εκτενώς στο βιβλίο [2] για αυτόματη επεξεργασία φυσικής γλώσσας. Η διαδικασία ξεκινά με την αυτόματη διόρθωση ορθογραφίας μέσω της συνάρτησης `sentence.correct()`, συνεχίζεται με την εξαγωγή ετικετών POS και ονοματικών φράσεων και ακολουθεί η επιφανειακή αναδιοργάνωση της πρότασης σε δομή S-V-O. Στο τελικό στάδιο πραγματοποιείται καθαρισμός και μορφοποίηση του κειμένου, ώστε να παραχθεί ένα ενιαίο και αναγνώσιμο αποτέλεσμα. Η προσέγγιση αυτή χαρακτηρίζεται από αυτόματη ορθογραφική διόρθωση, χρήση στατιστικού μοντέλου βασισμένου σε corpus, καθώς και από γρήγορη εκτέλεση με χαμηλές απαιτήσεις σε υπολογιστικούς πόρους.

Η κεντρική συνάρτηση που κατευθύνει τη δημιουργία του ανακατασκευασμένου κειμένου είναι η `reconstruct_text_with_textblob`. Η συνάρτηση αυτή δημιουργεί ένα αντικείμενο TextBlob και αξιοποιεί τον αυτόματο μηχανισμό sentence segmentation της βιβλιοθήκης, εφαρμόζοντας την ανακατασκευή σε επίπεδο μεμονωμένης πρότασης, ώστε να διατηρείται η τοπική συνοχή του κειμένου. Αφού ολοκληρωθεί η επεξεργασία κάθε πρότασης, οι προτάσεις ενώνονται μεταξύ τους με κενά πριν επιστραφούν στη main συνάρτηση, με αποτέλεσμα την παραγωγή ενός ενιαίου κειμένου.

Η ανακατασκευή σε επίπεδο πρότασης υλοποιείται στη συνάρτηση `_reconstruct_sentence()`, όπου εφαρμόζεται αρχικά η ορθογραφική διόρθωση από το TextBlob, ακολουθεί η εξαγωγή γλωσσικών χαρακτηριστικών και στη συνέχεια πραγματοποιείται επιφανειακή αναδιοργάνωση της πρότασης με βάση τα στατιστικά χαρακτηριστικά που παρέχει η βιβλιοθήκη. Στη συνάρτηση `_reorganize_by_pos` υλοποιείται μια επιφανειακή ομαδοποίηση των tokens με βάση τις ετικέτες POS, με στόχο την απλοποίηση και εξομάλυνση της πρότασης χωρίς πλήρη συντακτική αναδόμηση. Η διαδικασία αυτή λειτουργεί ως heuristic sentence simplification και ευθυγραμμίζεται με επιφανειακές τεχνικές NLP, όπως περιγράφονται στο βιβλίο [1]. Τέλος, εφαρμόζεται string-level post-processing στο ανακατασκευασμένο κείμενο, με αντίστοιχη λογική με το Α ερώτημα, για την τελική μορφοποίηση και καθαρισμό της εξόδου.

Pipeline 2: NLTK + Embeddings

Το Pipeline 2 υιοθετεί μια σημασιολογική προσέγγιση για την ανακατασκευή κειμένου, βασισμένη στη χρήση προεκπαιδευμένων word embeddings και συγκεκριμένα των GloVe embeddings. Το μοντέλο που αξιοποιείται είναι το `glove-wiki-gigaword-100`, ένα διανυσματικό μοντέλο 100 διαστάσεων εκπαιδευμένο σε μεγάλο σώμα κειμένων από τη Wikipedia, το οποίο επιτρέπει την αποτύπωση σημασιολογικών σχέσεων μεταξύ λέξεων. Στόχος της προσέγγισης είναι η διατήρηση του αρχικού νοήματος του κειμένου, επιτυγχάνοντας ταυτόχρονα μια διαφορετική, αλλά σημασιολογικά ισοδύναμη διατύπωση.

Η διαδικασία ξεκινά με tokenization και Part-of-Speech tagging, ώστε να εντοπιστούν με ακρίβεια οι λέξεις περιεχομένου, δηλαδή ουσιαστικά, ρήματα, επίθετα και επιρρήματα. Για κάθε μία από αυτές τις λέξεις αναζητούνται σημασιολογικά παρόμοιες λέξεις στον διανυσματικό χώρο των embeddings, με βάση τον υπολογισμό της cosine similarity, η οποία εκφράζει τον βαθμό σημασιολογικής εγγύτητας μεταξύ δύο διανυσμάτων. Η αντικατάσταση μιας λέξης πραγματοποιείται μόνο όταν η τιμή της ομοιότητας υπερβαίνει ένα προκαθορισμένο κατώφλι ίσο με 0.65, ώστε να ελαχιστοποιείται ο κίνδυνος αλλοίωσης του νοήματος.

Αφού ολοκληρωθεί η διαδικασία αντικατάστασης, οι λέξεις επανασυντίθενται σε προτάσεις, οδηγώντας στην τελική μορφή του ανακατασκευασμένου κειμένου. Η συγκεκριμένη προσέγγιση δεν βασίζεται σε ρητούς συντακτικούς ή γραμματικούς κανόνες, αλλά αποκλειστικά στη σημασιολογική πληροφορία που ενσωματώνεται στα embeddings. Ως αποτέλεσμα, το pipeline προσφέρει σημασιολογική κατανόηση μέσω διανυσματικών αναπαραστάσεων, δυνατότητα εύρεσης συνωνύμων και σχετικών λέξεων και διατήρηση του αρχικού νοήματος με εναλλακτική διατύπωση.

Pipeline 3: Transformer

Το Pipeline 3 υιοθετεί μια νευρωνική προσέγγιση για την ανακατασκευή κειμένου και διατυπώνει το πρόβλημα ως διαδικασία text-to-text generation, αξιοποιώντας ένα προεκπαιδευμένο encoder–decoder transformer μοντέλο. Συγκεκριμένα, χρησιμοποιείται το μοντέλο `google/flan-t5-base`, μεγέθους περίπου 250 MB, το οποίο έχει εκπαιδευτεί για γενικές εργασίες μετασχηματισμού κειμένου και επιτρέπει την παραγωγή βελτιωμένων, σαφών και γραμματικά ορθών εκδοχών ενός δοθέντος κειμένου. Το αρχικό κείμενο παρέχεται απευθείας στο μοντέλο μέσω κατάλληλου prompt, το οποίο καθοδηγεί τη διαδικασία αναδιατύπωσης με στόχο τη διόρθωση γραμματικών λαθών και τη διαμόρφωση επίσημου και κατανοητού ύφους.

Η αρχιτεκτονική του μοντέλου βασίζεται στη διάκριση encoder και decoder. Ο encoder επεξεργάζεται την είσοδο χρησιμοποιώντας μηχανισμούς self-attention, παράγοντας συμφραζόμενες αναπαραστάσεις που αποτυπώνουν τη γραμματική, τη σύνταξη και το συνολικό νόημα του κειμένου. Στη συνέχεια, ο decoder αξιοποιεί cross-attention πάνω στις αναπαραστάσεις του encoder και δημιουργεί τη νέα εκδοχή του κειμένου token-by-token. Η παραγωγή της εξόδου ελέγχεται μέσω τεχνικών όπως beam search ή sampling, με παραμέτρους που ρυθμίζουν το μέγιστο μήκος της εξόδου, τον βαθμό τυχαιότητας στην επιλογή tokens, το nucleus sampling και την αποφυγή επαναλήψεων, ώστε να επιτυγχάνεται ισορροπία μεταξύ ποιότητας και ποικιλίας στην παραγόμενη έξοδο.

Η συγκεκριμένη προσέγγιση δεν βασίζεται σε χειροκίνητους κανόνες, POS patterns ή επιφανειακές αντικαταστάσεις λέξεων, αλλά αξιοποιεί ενδογενώς μαθημένες γλωσσικές και σημασιολογικές σχέσεις, προσφέροντας βαθιά κατανόηση γραμματικής και σύνταξης, καθώς και δυνατότητα παράφρασης και αναδιατύπωσης. Ωστόσο, συνοδεύεται από αυξημένες υπολογιστικές απαιτήσεις σε σχέση με τις προηγούμενες προσεγγίσεις. Τέλος, μετά την παραγωγή του κειμένου εφαρμόζεται μόνο ελάχιστο post-processing, περιορισμένο κυρίως στη μορφοποίηση της τελικής εξόδου, καθώς το μεγαλύτερο μέρος της βελτίωσης επιτυγχάνεται απευθείας από το νευρωνικό μοντέλο.

Πειράματα και Αποτελέσματα

Παραδοτέο 1Α

Preprocessing.py

Το στάδιο της προεπεξεργασίας εφαρμόστηκε στις δύο επιλεγμένες προτάσεις με στόχο την κανονικοποίηση και προετοιμασία τους για τα επόμενα στάδια ανάλυσης. Η πρώτη πρόταση, υποβλήθηκε σε επτά διαδοχικά βήματα επεξεργασίας. Αρχικά δεν απαιτήθηκε επέκταση συντομογραφιών καθώς δεν υπήρχαν contractions στο κείμενο. Ακολούθησε η μετατροπή σε πεζά γράμματα και η αφαίρεση των σημείων στίξης, με αποτέλεσμα την παραγωγή ενός καθαρού αλφαριθμητικού. Το tokenization με χρήση NLTK παρήγαγε 19 tokens, ενώ το POS tagging ανέθεσε γραμματικές ετικέτες σε κάθε token, εντοπίζοντας 6 ουσιαστικά, 3 προθέσεις, 2 κτητικές αντωνυμίες και διάφορα ρήματα. Η λημματοποίηση μετέτρεψε τις λέξεις στη βασική τους μορφή, όπως το "is" σε "be".

Η δεύτερη πρόταση, που αναφερόταν σε υποβολή εργασίας και ενημερώσεις, ακολούθησε την ίδια διαδικασία επεξεργασίας. Λόγω του μεγαλύτερου μήκους και της πιο σύνθετης δομής της, το tokenization παρήγαγε 35 tokens. Το POS tagging εντόπισε 8 ουσιαστικά, 5 προθέσεις, 3 ρήματα σε παρελθοντικό χρόνο και 3 επιρρήματα, αποκαλύπτοντας μια πιο περίπλοκη συντακτική δομή σε σχέση με την πρώτη πρόταση.

Syntactical_analysis.py

Η συντακτική ανάλυση της πρώτης πρότασης εντόπισε τρεις κύριες ονοματικές φράσεις: "our dragon boat festival", "our Chinese culture" και "our lives". Οι ρηματικές ομάδες που αναγνωρίστηκαν περιλάμβαναν το κύριο ρήμα "is" και το απαρεμφατικό "to celebrate". Η εξαγωγή της δομής SVO προσδιόρισε ως υποκείμενο το "Today", ως ρήμα το "is" και ως αντικείμενο το "our dragon boat festival". Η δομή των προτάσεων αναλύθηκε σε κύρια πρόταση, προθετική φράση και απαρεμφατική πρόταση, οδηγώντας σε μια ανακατασκευασμένη εκδοχή με βελτιωμένη οργάνωση.

Η συντακτική ανάλυση της δεύτερης πρότασης αποκάλυψε σημαντικά περισσότερες ονοματικές φράσεις, συμπεριλαμβανομένων των "the new submission", "last autumn", "the updates" και "the full feedback". Εντοπίστηκαν πολλαπλές ρηματικές ομάδες όπως "told", "were waiting" και "was confusing", με την τελευταία να παρουσιάζει πρόβλημα συμφωνίας υποκειμένου-ρήματος. Η ανάλυση εντόπισε επίσης δύο δευτερεύουσες προτάσεις και το ελλιπές βοηθητικό ρήμα στη φράση "it not included". Η εξαγωγή SVO προσδιόρισε ως υποκείμενο το "I", ως ρήμα το "told" και ως αντικείμενο το "him".

Grammatical_correction.py

Η γραμματική διόρθωση της πρώτης πρότασης δεν απαίτησε σημαντικές αλλαγές καθώς η συμφωνία υποκειμένου-ρήματος ήταν ήδη σωστή με το "Today is", η μορφολογική συνέπεια διατηρούνταν με ενιαίο ενεστώτα χρόνο και η συμφωνία άρθρου-ουσιαστικού ήταν ορθή. Το post-processing περιορίστηκε στην προσθήκη κεφαλαίου στο πρώτο γράμμα και τελείας στο τέλος της πρότασης.

Η δεύτερη πρόταση απαίτησε εκτεταμένη γραμματική διόρθωση. Η φράση "the updates was confusing" διορθώθηκε σε "the updates were confusing" για να επιτευχθεί σωστή συμφωνία υποκειμένου-ρήματος. Η ελλιπής δομή "it not included" συμπληρώθηκε με το βοηθητικό ρήμα σε "it did not include". Επιπλέον, η λέξη "discuss" αντικαταστάθηκε με την ουσιαστική μορφή "discussion" και η φράση "were waiting" βελτιώθηκε σε "had been waiting" για καλύτερη χρονική ακολουθία. Το τελικό αποτέλεσμα ήταν μια γραμματικά ορθή και συνεκτική πρόταση.

Παραδοτέο 1Β – TextBlob

Το πρώτο pipeline εφάρμοσε στατιστική επεξεργασία με τη χρήση της βιβλιοθήκης TextBlob στα δύο κείμενα εισόδου. Η διαδικασία ξεκίνησε με τον αυτόματο διαχωρισμό κάθε κειμένου σε προτάσεις, εντοπίζοντας 6 προτάσεις στο πρώτο κείμενο και 6 στο δεύτερο. Η ορθογραφική διόρθωση του TextBlob δεν πραγματοποίησε σημαντικές αλλαγές καθώς οι περισσότερες λέξεις ήταν ορθογραφικά σωστές, αν και γραμματικά προβληματικές.

Η εξαγωγή POS tags αναγνώρισε τις κατηγορίες λέξεων σε κάθε πρόταση. Στο πρώτο κείμενο εντοπίστηκαν ουσιαστικά όπως "festival", "culture", "wishes" και "message", ρήματα όπως "celebrate", "hope", "enjoy" και "received", καθώς και επίθετα όπως "dragon", "Chinese", "safe" και "great". Η αναδιοργάνωση με βάση το pattern SVO εφαρμόστηκε επιφανειακά σε κάθε πρόταση, ομαδοποιώντας τα tokens ανά γραμματική κατηγορία.

Το pipeline ολοκληρώθηκε σε λιγότερο από 1 δευτερόλεπτο με ελάχιστη χρήση μνήμης περίπου 50 MB. Τα αποτελέσματα ήταν προβλέψιμα και σταθερά μεταξύ εκτελέσεων. Ωστόσο, το pipeline δεν διόρθωσε τα γραμματικά λάθη όπως "the updates was" ή "it not included", περιορίζοντας την αποτελεσματικότητά του σε επιφανειακή μορφοποίηση και αναδιοργάνωση χωρίς ουσιαστική γραμματική βελτίωση.

Παραδοτέο 1Β – Embeddings

Το δεύτερο pipeline υιοθέτησε σημασιολογική προσέγγιση χρησιμοποιώντας τα προεκπαιδευμένα GloVe embeddings με 100 διαστάσεις. Η φόρτωση του μοντέλου glove-wiki-gigaword-100 απαίτησε 5-10 δευτερόλεπτα κατά την πρώτη εκτέλεση, με τα embeddings να αποθηκεύονται στη μνήμη για επόμενες χρήσεις.

Η διαδικασία ξεκίνησε με τον εντοπισμό των λέξεων περιεχομένου μέσω POS tagging, επικεντρώνοντας σε ουσιαστικά, ρήματα, επίθετα και επιρρήματα. Για κάθε λέξη περιεχομένου αναζητήθηκαν σημασιολογικά παρόμοιες λέξεις στον διανυσματικό χώρο των embeddings.

Το αποτέλεσμα ήταν κείμενα με αυξημένη λεξιλογική ποικιλία που διατηρούσαν το γενικό νόημα αλλά με διαφορετική διατύπωση. Ωστόσο, τα γραμματικά λάθη παρέμειναν αδιόρθωτα καθώς το pipeline δεν εφαρμόζει γραμματικούς κανόνες. Η χρήση μνήμης ανήλθε σε περίπου 200 MB λόγω της φόρτωσης των embeddings.

Παραδοτέο 1Β – Transformer

Το τρίτο pipeline αξιοποίησε το προεκπαιδευμένο μοντέλο google/flan-t5-base, ένα encoder-decoder transformer μεγέθους περίπου 250 MB. Η επεξεργασία διατυπώθηκε ως πρόβλημα text-to-text generation, όπου το μοντέλο λαμβάνει το αρχικό κείμενο μαζί με ένα prompt που το καθοδηγεί να διορθώσει γραμματικά λάθη και να βελτιώσει τη σαφήνεια και τον επίσημο τόνο.

Οι παράμετροι generation ρυθμίστηκαν για βέλτιστη ισορροπία μεταξύ ποιότητας και ποικιλίας: μέγιστο μήκος 512 tokens, temperature 0.8 για ελεγχόμενη τυχαιότητα, top_p 0.95 για nucleus sampling και repetition_penalty 1.2 για αποφυγή επαναλήψεων. Η εκτέλεση πραγματοποιήθηκε σε CPU, απαιτώντας 10-30 δευτερόλεπτα ανά κείμενο.

Στο πρώτο κείμενο, το μοντέλο πραγματοποίησε εκτεταμένες διορθώσεις: η φράση "to celebrate it with all safe and great" αναδιατυπώθηκε σε πιο φυσική μορφή, το "Hope you too, to enjoy" διορθώθηκε σε "I hope you also enjoy" με προσθήκη υποκειμένου, το "Thank your message" συμπληρώθηκε σε "Thank you for your message" και το "I am very appreciated" διορθώθηκε στο ορθό "I greatly appreciate".

Στο δεύτερο κείμενο, οι διορθώσεις ήταν ακόμη πιο εκτεταμένες. Η φράση "final discuss" έγινε "final discussion", το "the updates was confusing" διορθώθηκε σε "the updates were confusing", το "it not included" συμπληρώθηκε με βοηθητικό ρήμα σε "it did not include", το

"although bit delay" αναδιατυπώθηκε σε "although there was a bit of delay", το "they really tried best" έγινε "they really tried their best", το "if the doctor still plan" διορθώθηκε σε "if the doctor still plans" και το "before he sending" σε "before he sends".

Το pipeline παρήγαγε κείμενα με υψηλή γραμματική ορθότητα, φυσική ροή και επίσημο ύφος. Ωστόσο, η υψηλή χρήση μνήμης (περίπου 500 MB) και ο αυξημένος χρόνος εκτέλεσης αποτελούν πρακτικούς περιορισμούς. Επιπλέον, λόγω του stochastic sampling, τα αποτελέσματα μπορεί να διαφέρουν ελαφρώς μεταξύ εκτελέσεων.

Συζήτηση

Η συζήτηση των αποτελεσμάτων ανέδειξε ουσιαστικές διαφορές μεταξύ των προσεγγίσεων ανακατασκευής κειμένου, τόσο ως προς το επίπεδο ανάλυσης όσο και ως προς την ποιότητα της παραγόμενης εξόδου. Οι rule-based μέθοδοι αποδείχθηκαν αποτελεσματικές στη διόρθωση συγκεκριμένων και προβλέψιμων γραμματικών σφαλμάτων, ωστόσο η απόδοσή τους περιορίστηκε σημαντικά από την απουσία μηχανισμών σημασιολογικής κατανόησης. Η αποτελεσματικότητά τους εξαρτήθηκε σε μεγάλο βαθμό από τη δομή και τη σαφήνεια της αρχικής πρότασης, γεγονός που τις καθιστά λιγότερο αξιόπιστες σε περιπτώσεις ασάφειας, ιδιωματικών εκφράσεων ή σύνθετης σύνταξης. Αντίστοιχα, οι στατιστικές προσεγγίσεις, όπως το TextBlob, παρείχαν γρήγορα και προβλέψιμα αποτελέσματα για βασικές διορθώσεις, χωρίς όμως να υποστηρίζουν βαθιά κατανόηση του περιεχομένου ή των συμφραζομένων.

Η αξιοποίηση word embeddings (GloVe) επέτρεψε τη σημασιολογική ανάλυση και την εύρεση συνωνύμων ή σχετικών εννοιών μέσω ποσοτικής μέτρησης ομοιότητας στον διανυσματικό χώρο. Οι προτάσεις που προέκυψαν εμφάνισαν αυξημένη λεξιλογική ποικιλία και μεγαλύτερη ευελιξία σε σύγκριση με τις rule-based προσεγγίσεις, γεγονός που υποδηλώνει καλύτερη αποτύπωση σημασιολογικών σχέσεων. Παρ’ όλα αυτά, η έλλειψη κατανόησης συμφραζομένων, η αδυναμία χειρισμού πολυσημίας και η εξάρτηση από το εκάστοτε training corpus οδήγησαν σε περιπτώσεις αλλοίωσης του αρχικού νοήματος ή συντακτικών ασυνεπειών. Τα ευρήματα αυτά καταδεικνύουν ότι η σημασιολογική ομοιότητα σε επίπεδο λέξεων δεν επαρκεί από μόνη της για ολοκληρωμένη και συνεπή ανακατασκευή προτάσεων.

Η transformer-based προσέγγιση παρουσίασε την υψηλότερη ποιότητα ανακατασκευής, με σαφή υπεροχή ως προς τη γραμματική ορθότητα, τη συνοχή και τη φυσικότητα της παραγόμενης γλώσσας. Μέσω του μηχανισμού προσοχής, το μοντέλο κατόρθωσε να λαμβάνει υπόψη μακρινές εξαρτήσεις και σύνθετα συμφραζόμενα, επιτυγχάνοντας αναδιατύπωση σε επίπεδο πρότασης και όχι μεμονωμένων λέξεων. Ωστόσο, η αυξημένη υπολογιστική πολυπλοκότητα και η μειωμένη προβλεψιμότητα της εξόδου αποτελούν πρακτικούς περιορισμούς. Συνολικά, τα αποτελέσματα επιβεβαιώνουν μια σαφή μετάβαση από επιφανειακή σε συμφραζόμενη επεξεργασία φυσικής γλώσσας, όπου τα πιο σύνθετα νευρωνικά μοντέλα οδηγούν σε ποιοτικότερη ανακατασκευή, με το αντίστοιχο όμως κόστος σε υπολογιστικούς πόρους.

Τέλος, η μελέτη παρουσιάζει ορισμένους περιορισμούς που πρέπει να ληφθούν υπόψη. Το περιορισμένο σύνολο δεδομένων δεν επιτρέπει τη γενίκευση των συμπερασμάτων, ενώ δεν πραγματοποιήθηκε fine-tuning των μοντέλων ούτε εκτενής βελτιστοποίηση των παραμέτρων των pipelines. Η αξιολόγηση βασίστηκε κυρίως σε ποιοτική ανάλυση και σημασιολογική ομοιότητα, χωρίς ανθρώπινη κρίση μεγάλης κλίμακας. Παρά τους περιορισμούς αυτούς, τα αποτελέσματα υποδεικνύουν ότι μια συνδυαστική προσέγγιση, με preprocessing βασισμένο σε

κανόνες, ανακατασκευή μέσω neural μοντέλων και post-processing με γραμματικούς ελέγχους, αποτελεί μια πολλά υποσχόμενη κατεύθυνση για μελλοντική εργασία.

Συμπεράσματα

Η σύγκριση των διαφορετικών προσεγγίσεων ανέδειξε μια σαφή μετάβαση από επιφανειακές, κανονοκεντρικές μεθόδους σε πιο σύνθετες και συμφραζόμενες τεχνικές επεξεργασίας φυσικής γλώσσας. Οι rule-based προσεγγίσεις αποδείχθηκαν αποτελεσματικές για συγκεκριμένα και προβλέψιμα γραμματικά σφάλματα, αλλά περιορίζονται σε προκαθορισμένα patterns. Οι στατιστικές μέθοδοι, όπως το TextBlob, προσφέρουν γρήγορες και αξιόπιστες βασικές διορθώσεις χωρίς όμως βαθιά κατανόηση του κειμένου. Τα semantic embeddings, όπως τα GloVe, επιτρέπουν ανάλυση σημασιολογικής ομοιότητας και εύρεση εναλλακτικών διατυπώσεων, με τον κίνδυνο ωστόσο αλλοίωσης του αρχικού νοήματος. Αντίθετα, τα νευρωνικά transformer μοντέλα, όπως το Flan-T5, παρείχαν τη μεγαλύτερη ευελιξία και ποιότητα ανακατασκευής, αξιοποιώντας βαθιά κατανόηση γραμματικής, σύνταξης και συμφραζομένων, με κόστος αυξημένες υπολογιστικές απαιτήσεις.

Παρά τα θετικά αποτελέσματα, αναδείχθηκαν σημαντικές προκλήσεις, όπως η επίτευξη ισορροπίας μεταξύ γραμματικής διόρθωσης και διατήρησης του αρχικού νοήματος, η διαχείριση πολύγλωσσων κειμένων και η κατανόηση υφολογικών αποχρώσεων. Μελλοντικές κατευθύνσεις περιλαμβάνουν το fine-tuning transformer μοντέλων σε εξειδικευμένα σύνολα δεδομένων, τον συνδυασμό rule-based και νευρωνικών προσεγγίσεων για υβριδικά συστήματα, καθώς και την αξιολόγηση της ποιότητας της ανακατασκευής με τη συνδρομή ανθρώπινης κρίσης. Συνολικά, τα ευρήματα επιβεβαιώνουν ότι η αύξηση της πολυπλοκότητας των μοντέλων οδηγεί σε ποιοτικότερη ανακατασκευή κειμένου, με αντίστοιχο όμως υπολογιστικό κόστος.

Βιβλιογραφία

[1] A. Kulkarni and A. Shivananda, “Natural Language Processing Recipes: Unlocking Text Data with Machine Learning and Deep Learning using Python,” Apress, 2019.

[2] S. Bird, E. Klein, and E. Loper, “Natural Language Processing with Python,” O’Reilly Media, 2009.

[3] TextBlob Documentation. https://textblob.readthedocs.io/

[4] NLTK Documentation. https://www.nltk.org/

[5] Hugging Face Transformers. https://huggingface.co/docs/transformers/

[6] Gensim Documentation. https://radimrehurek.com/gensim/

[7] Vaswani, A., et al. “Attention Is All You Need.” NeurIPS 2017.