



# ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

## Διόρθωση κειμένων με χρήση τεχνικών ενσωμάτωσης φυσικής γλώσσας



ΚΑΠΕΛΛΕΣ ΔΙΟΝΥΣΙΟΣ

Οκτώβριος 2024



Διόρθωση κειμένων με χρήση τεχνικών ενσωμάτωσης φυσικής γλώσσας

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ ΤΟΥ  
ΚΑΠΕΛΛΕ ΔΙΟΝΥΣΙΟΥ  
Α.Μ: 1067479

Τριμελής Επιτροπή:

Δ. Κουτσομητρόπουλος, μέλος ΕΔΙΠ (επιβλέπων)

Χ. Μακρής, Αν. Καθηγητής

Σ. Σιούτας, Καθηγητής

## *EΥΧΑΡΙΣΤΙΕΣ*

Θα ήθελα να ευχαριστήσω από καρδίας τους επιβλέποντες καθηγητές μου κύριους Κουτσομητρόπουλο Δημήτριο και Ανδριόπουλο Ανδρέα για την εμπιστοσύνη που μου έδειξαν κατά την διεκπεραίωση της παρούσας διπλωματικής εργασίας, για τις συμβουλές και για την στήριξη που μου παρείχαν όλο αυτό το διάστημα τόσο σε επαγγελματικό όσο και σε προσωπικό επίπεδο.

Ακόμη, είναι σημαντικό να ευχαριστήσω το Τμήμα Μηχανικών Η/Υ και Πληροφορικής του Πανεπιστημίου Πατρών για τις γνώσεις που μου παρείχαν ώστε να καταφέρω να διεκπεραιώσω την παρούσα διπλωματική.

Τέλος, θα ήθελα να εκφράσω ένα τεράστιο ευχαριστώ στην οικογένεια και στους φίλους μου για την στήριξη και την αγάπη που μου έδειξαν καθ' όλη την διάρκεια των σπουδών μου. Χωρίς την συμπαράσταση τους, όλο αυτό το όμορφο ταξίδι θα ήταν πολύ πιο δύσκολο.

## Περίληψη

Η αυτόματη διόρθωση γραμματικών λαθών (Grammatical Error Correction, GEC) περιλαμβάνει την αυτόματη διόρθωση διαφόρων τύπων γραμματικών σφαλμάτων, συμπεριλαμβανομένων της ορθογραφίας, των σημείων στίξης και της γραμματικής. Προκειμένου να μετατραπεί μια λανθασμένη πρόταση στη σωστή της έκδοση, ένα σύστημα GEC απαιτεί συνήθως της εισαγωγή της ίδιας της πρότασης. Υπάρχουν πολλές προσεγγίσεις για τη διόρθωση γραμματικών λαθών, που κυμαίνονται από μοντέλα βασισμένα σε κανόνες έως νευρωνική μηχανική μετάφραση. Αυτή η διατριβή εμβαθύνει στον τομέα της επεξεργασίας φυσικής γλώσσας διερευνώντας τη λεπτομερή ρύθμιση του μοντέλου T5 που βασίζεται σε Transformer (Text-to-Text Transfer Transformer) για τη συγκεκριμένη εργασία της γραμματικής διόρθωσης. Η ακριβής γραμματική διόρθωση είναι πρωταρχικής σημασίας για την αποτελεσματική επικοινωνία, ειδικά για τους μη μητρικούς ομιλητές μιας γλώσσας. Η έρευνα στοχεύει να αξιοποιήσει τη δύναμη των παραγωγικών δυνατοτήτων του μοντέλου T5 και να μεταφέρει τη μάθηση για την ανάπτυξη ενός αποτελεσματικού και ευέλικτου συστήματος για αυτοματοποιημένη γραμματική διόρθωση σε γραπτό κείμενο. Περιλαμβάνει τη λεπτομέρεια ενός προεκπαιδευμένου μοντέλου T5 σε ένα προσαρμοσμένο σύνολο δεδομένων που περιέχει προτάσεις με διάφορους βαθμούς γραμματικών σφαλμάτων. Η προεπεξεργασία δεδομένων περιλαμβάνει την κωδικοποίηση των προτάσεων στη μορφή T5, επιτρέποντάς του να δημιουργεί διορθωμένες προτάσεις για εισαγωγή με γραμματικά λάθη. Τα αποτελέσματα δείχνουν την αποτελεσματικότητα του βελτιωμένου μοντέλου T5 στη γραμματική διόρθωση. Το μοντέλο επιτυγχάνει ανταγωνιστικές επιδόσεις σε μετρήσεις συγκριτικής αξιολόγησης, ξεπερνώντας τις υπάρχουσες μεθόδους όσον αφορά την ακρίβεια και την κατανόηση των συμφραζομένων. Τα ευρήματα υπογραμμίζουν τη σημασία της χρήσης προεκπαιδευμένων μοντέλων και τεχνικών λεπτομέρειας για την ανάπτυξη εξελιγμένων συστημάτων διόρθωσης γραμματικής. Συμπερασματικά, αυτή η μελέτη υπογραμμίζει την ικανότητα ενός βελτιωμένου μοντέλου T5 στην αντιμετώπιση των προκλήσεων της γραμματικής διόρθωσης. Οι γνώσεις που αποκτήθηκαν ανοίγουν το δρόμο για μελλοντική έρευνα για τη βελτίωση των αρχιτεκτονικών μοντέλων και την επέκταση του πεδίου εφαρμογής των αυτοματοποιημένων εφαρμογών διόρθωσης γλώσσας.

**Λέξεις κλειδιά:** GEC, Προεκπαιδευμένο Μοντέλο, T5

## Abstract

Grammatical Error Correction (GEC) involves the automatic correction of various types of grammatical errors, including spelling, punctuation and grammar. In order to convert an incorrect sentence to the correct version, a GEC system usually requires the input of the sentence itself. There are many approaches to grammatical error correction, ranging from rule-based models to neural machine translation. This thesis delves into the field of natural language processing by exploring the detailed setup of the T5-based Text-to-Text Transfer Transformer (T5) model for the specific task of grammatical correction. Accurate grammatical correction is paramount for effective communication, especially for non-native speakers of a language. This research aims to harness the power of the productive capabilities of the T5 model and transfer learning to develop an efficient and flexible system for automated grammatical correction in written text. It involves the detail of a pre-trained T5 model on a custom dataset containing sentences with varying degrees of grammatical errors. Data preprocessing involves encoding the sentences in the T5 format, allowing it to generate corrected sentences for input with grammatical errors. The results show the effectiveness of the improved T5 model in grammatical correction. The model achieves competitive performance on benchmarking metrics, outperforming existing methods in terms of accuracy and contextual understanding. The findings highlight the importance of using pre-trained models and detail techniques to develop sophisticated grammar correction systems. In conclusion, this study highlights the ability of an improved T5 model in addressing the challenges of grammar correction. The insights gained pave the way for future research to improve architectural models and expand the scope of automated language correction applications.

**Keywords:** GEC, Pretrained Model, T5

## Πίνακας περιεχομένων

<b>1. Εισαγωγή .....</b>	<b>7</b>
<b>1.1 Στόχος Διπλωματικής.....</b>	<b>8</b>
<b>2. Background της GEC .....</b>	<b>9</b>
<b>2.1 Προσεγγίσεις της GEC .....</b>	<b>10</b>
<b>Rule-based systems.....</b>	<b>11</b>
<b>Classification-based systems.....</b>	<b>13</b>
<b>Machine translation .....</b>	<b>16</b>
<b>Statistical Machine Translation .....</b>	<b>16</b>
<b>Neural Machine Translation .....</b>	<b>18</b>
<b>2.2 Διαμοιρασμένες Εργασίες .....</b>	<b>20</b>
<b>CoNLL-2014.....</b>	<b>21</b>
<b>BEA-2019 .....</b>	<b>27</b>
<b>2.3 Σχετικές Εργασίες .....</b>	<b>33</b>
<b>Junczys-Dowmunt (2018).....</b>	<b>34</b>
<b>Zhao (2019).....</b>	<b>36</b>
<b>Kiyono (2019).....</b>	<b>38</b>
<b>Kaneko (2020).....</b>	<b>40</b>
<b>Omelianchuk (2020) .....</b>	<b>42</b>
<b>Rothe (2021) .....</b>	<b>43</b>
<b>3. Μεθοδολογία Εκπαίδευσης Μοντέλου .....</b>	<b>45</b>
<b>3.1 Συμβατά Dataset .....</b>	<b>45</b>
<b>3.2 Ανάλυση Λειτουργίας Μοντέλου .....</b>	<b>47</b>
<b>3.3 Επιλογή Dataset .....</b>	<b>53</b>
<b>3.4 Προεπεξεργασία Δεδομένων .....</b>	<b>55</b>
<b>3.5 Εκπαίδευση Μοντέλου .....</b>	<b>58</b>
<b>3.6 Δοκιμή Μοντέλου με Παραδείγματα .....</b>	<b>61</b>
<b>3.6.1 Παράδειγμα διόρθωσης ορθογραφίας σε κείμενο .....</b>	<b>64</b>
<b>3.6.2 Παράδειγμα διόρθωσης γραμματικής σε κείμενο .....</b>	<b>65</b>
<b>3.6.3 Παράδειγμα διόρθωσης σημείων στίξης σε κείμενο .....</b>	<b>65</b>
<b>3.6.4 Παράδειγμα διόρθωσης πολλαπλών λαθών σε κείμενο .....</b>	<b>66</b>
<b>3.6.5 Παράδειγμα αναγνώρισης σωστού κειμένου .....</b>	<b>66</b>
<b>4. Χρήση μοντέλου σε ελληνικό dataset .....</b>	<b>68</b>
<b>4.1 Επιλογή προ-εκπαίδευμένου μοντέλου.....</b>	<b>68</b>
<b>4.2 Λεπτομερής ρύθμιση και επιλογή dataset.....</b>	<b>68</b>
<b>4.3 Αξιολόγηση μοντέλου μέσω μετρικής ELERRANT.....</b>	<b>70</b>
<b>4.4 Χρήση και αξιολόγηση μοντέλου mT5 σε ελληνικό dataset .....</b>	<b>72</b>
<b>5. Αξιολόγηση Μοντέλου .....</b>	<b>74</b>
<b>5.1 Μετρικές Αξιολόγησης.....</b>	<b>74</b>
<b>5.2 Υπολογισμός Απώλειας Μοντέλου.....</b>	<b>78</b>
<b>5.3 Υπολογισμός GLEU score .....</b>	<b>83</b>
<b>5.4 Εφαρμογή της μετρικής GLEU στο σύστημα πριν το fine-tuning .....</b>	<b>88</b>
<b>5.5 Εφαρμογή της μετρικής GLEU στο σύστημα μετά το fine-tuning .....</b>	<b>90</b>
<b>5.6 Σύγκριση Αποτελεσμάτων στο GLEU benchmark .....</b>	<b>93</b>
<b>6. Συμπεράσματα.....</b>	<b>96</b>
<b>7. Μελλοντικές Εργασίες.....</b>	<b>97</b>
<b>Αναφορές.....</b>	<b>99</b>

## 1. Εισαγωγή

Η διόρθωση γραμματικών σφαλμάτων (Grammatical Error Correction, GEC) ορίζεται ως η επιδιόρθωση διαφόρων μορφών σφαλμάτων σε γραπτά έγγραφα, όπως ορθογραφικά, στίχης και γραμματικά προβλήματα. Η διόρθωση μιας λανθασμένης πρότασης απαιτεί συνήθως την απασχόληση ενός μηχανισμού που παίρνει τη λανθασμένη πρόταση και την αλλάζει στη σωστή της εκδοχή. Εφαρμογές GEC συναντώνται κυρίως σε επεξεργαστές κειμένου και σε διαδικτυακές υπηρεσίες γραφής, όπως το Grammarly [1]. Εκτός από την υποστήριξη του απλού χρήστη σε τακτικές γραπτές δραστηριότητες, τέτοια συστήματα μπορούν επίσης να βοηθήσουν τους μαθητές δεύτερης γλώσσας να βελτιώσουν τις δεξιότητές τους στη γλώσσα που επιθυμούν να μάθουν.

One option to moving toward both biodiversity and terrestrial food supply goals are to produce greater yield from less land.

GEC system

One option for moving toward both biodiversity and terrestrial food supply goals is to produce greater yields from less land.

Εικόνα 1.1: Ένα τυπικό παράδειγμα διόρθωσης γραμματικών λαθών

Οι δύο πιο πρόσφατες shared εργασίες, η CoNLL-2014 (Ng et al., 2014) και η BEA-2019 (Bryant, Felice, Andersen, & Briscoe, 2019), καταδεικνύουν την αυξανόμενη ελκυστικότητα του GEC ως θέμα Επεξεργασίας Φυσικής Γλώσσας ή NLP (Natural Language Processing). Οι δύο κοινές προκλήσεις απαιτούσαν από τους κατασκευαστές συστημάτων να δημιουργήσουν συστήματα GEC που θα διόρθωναν τις προτάσεις ενός πολυσυνόλου δεδομένων από διαφορετικές ομάδες μαθητών, με αποτέλεσμα ένα ευρύ φάσμα γραμματικών σφαλμάτων. Τα δύο συνεργατικά έργα όχι μόνο παρουσίασαν συστήματα GEC τελευταίας τεχνολογίας, αλλά και ανέδειξαν διάφορα ελαττώματα που εξακολουθούν να ταλαιπωρούν τα σύγχρονα συστήματα, όπως η διαχείριση προτάσεων με πολλαπλά λάθη.

Η παρούσα διπλωματική εργασία εμβαθύνει στον τομέα της επεξεργασίας φυσικής γλώσσας διερευνώντας τη λεπτομερή ρύθμιση του μοντέλου T5 ("Text-to-Text Transfer Transformer") για το συγκεκριμένο έργο της διόρθωσης γραμματικής, η οποία είναι υψίστης σημασίας για την αποτελεσματική επικοινωνία, ιδίως για τους μη φυσικούς ομιλητές μιας γλώσσας. Η έρευνα αποσκοπεί στην αξιοποίηση της δύναμης των γενετικών δυνατοτήτων του μοντέλου T5 για την ανάπτυξη ενός αποτελεσματικού και ευέλικτου συστήματος για την αυτοματοποιημένη διόρθωση της γραμματικής σε γραπτό κείμενο.

Εξετάζοντας την υπάρχουσα βιβλιογραφία εμβαθύνουμε στην εξέλιξη των αρχιτεκτονικών των νευρωνικών δικτύων, εστιάζοντας στην αρχιτεκτονική του μοντέλου T5 και στις εφαρμογές του σε διάφορες εργασίες επεξεργασίας φυσικής γλώσσας. Στη συνέχεια, εστιάζουμε στις ιδιαιτερότητες της διόρθωσης γραμματικής και την πρόσφατη τάση αξιοποίησης προ-εκπαιδευμένων γλωσσικών μοντέλων για το σκοπό αυτό.

Η προτεινόμενη μεθοδολογία περιλαμβάνει τη λεπτομερή ρύθμιση ενός προ-εκπαιδευμένου μοντέλου T5 σε ένα προσαρμοσμένο σύνολο δεδομένων που περιέχει προτάσεις με διαφορετικού βαθμού γραμματικά λάθη. Η προεπεξεργασία των δεδομένων περιλαμβάνει την κωδικοποίηση των προτάσεων στη μορφή T5. Το μοντέλο βελτιώνεται με τη λειτουργία sequence-to-sequence, επιτρέποντάς του να παράγει διορθωμένες προτάσεις για είσοδο με γραμματικά λάθη.

Τα πειραματικά αποτελέσματα αναδεικνύουν την αποτελεσματικότητα του λεπτομερώς ρυθμισμένου μοντέλου T5 στη γραμματική διόρθωση. Το μοντέλο επιτυγχάνει ανταγωνιστικές επιδόσεις στις μετρικές αξιολόγησης, ξεπερνώντας τις υπάρχουσες μεθόδους όσον αφορά την ακρίβεια και την κατανόηση του πλαισίου. Η μελέτη παρέχει πληροφορίες σχετικά με τα δυνατά σημεία του μοντέλου, την ικανότητά του να χειρίζεται ποικίλα μοτίβα σφαλμάτων και τους τομείς για περαιτέρω βελτίωση.

## 1.1 Στόχος Διπλωματικής

Η αποτελεσματική επικοινωνία μέσω του γραπτού λόγου είναι μια θεμελιώδης δεξιότητα στον σημερινό κόσμο της πληροφορίας. Ωστόσο, ακόμη και οι ικανοί συγγραφείς συχνά παλεύουν με γραμματικά λάθη που μπορούν να μειώσουν τη σαφήνεια και τον αντίκτυπο των μηνυμάτων τους. Στο πλαίσιο αυτό, οι τεχνολογίες επεξεργασίας φυσικής γλώσσας (NLP) έχουν αναδειχθεί ως ισχυρά εργαλεία για την αυτοματοποιημένη διαδικασία εντοπισμού και διόρθωσης αυτών των λαθών.

Η παρούσα έρευνα προσπαθεί να αξιοποιήσει τις δυνατότητες του μοντέλου T5 (Text-to-Text Transfer Transformer), ενός εξέχοντος μέλους της οικογένειας αρχιτεκτονικής Transformer, για να αντιμετωπίσει την πρόκληση της αυτοματοποιημένης διόρθωσης της γραμματικής. Οι εντυπωσιακές επιδόσεις του μοντέλου T5 σε ένα φάσμα εργασιών NLP, σε συνδυασμό με το πλαίσιο μετατροπής text-to-text, το καθιστούν έναν πολλά υποσχόμενο υποψήφιο για την επίτευξη αυτού του συγκεκριμένου στόχου.

Η κύρια εστίαση αυτής της διπλωματικής είναι στη διαδικασία τελειοποίησης (fine-tuning), κατά την οποία ένα προ-εκπαιδευμένο μοντέλο T5 προσαρμόζεται στο έργο της διόρθωσης γραμματικής. Ένα σχολαστικά επιμελημένο σύνολο δεδομένων, που περιλαμβάνει προτάσεις με ποικίλα γραμματικά λάθη, χρησιμεύει ως βάση για την εκπαίδευση του μοντέλου. Βήματα προεπεξεργασίας, όπως η κωδικοποίηση και η τροποποίηση των δεδομένων, προετοιμάζουν το σύνολο δεδομένων για την αποτελεσματική τελειοποίηση.

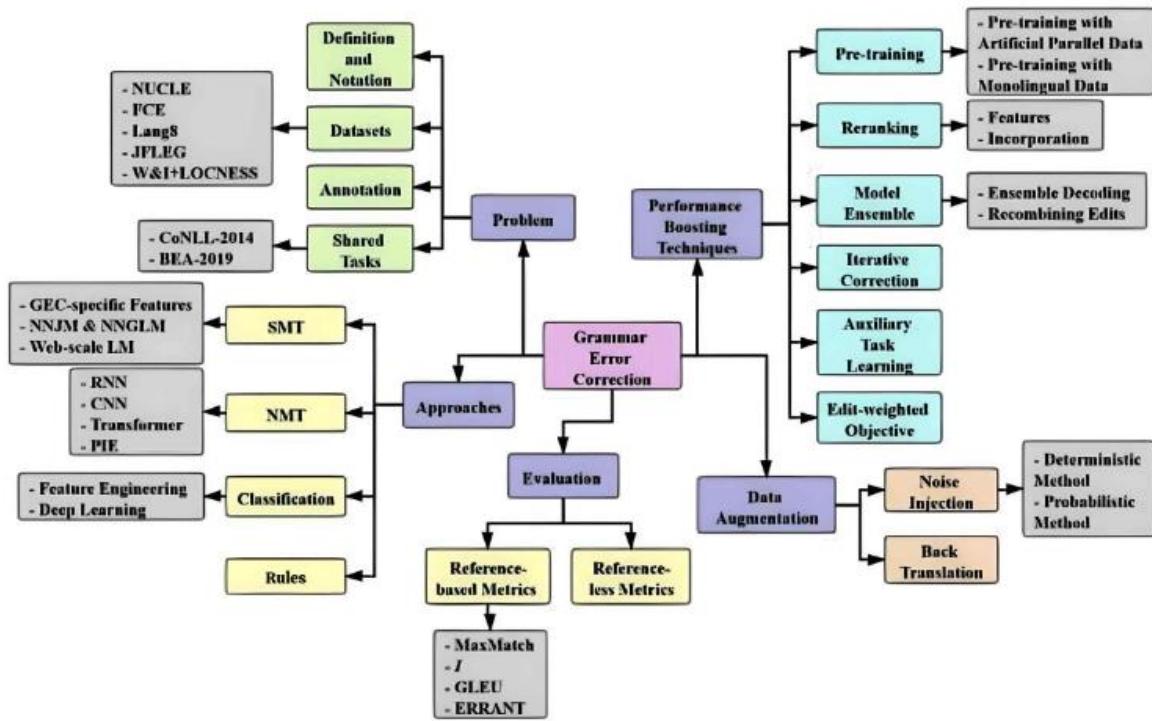
Τα πειραματικά αποτελέσματα καταδεικνύουν την αποτελεσματικότητα του μοντέλου T5 στη διόρθωση γραμματικής, καθώς υπερτερεί των κλασικών μεθόδων όσον αφορά την ακρίβεια και την επίγνωση των συμφραζόμενων, προσφέροντας διαφάνεια και παιδαγωγική αξία στους χρήστες.

Οι επιπτώσεις αυτής της έρευνας επεκτείνονται σε ένα ευρύ φάσμα ενδιαφερομένων, συμπεριλαμβανομένων των γλωσσικών μαθητών, των δημιουργών περιεχομένου, των εκπαιδευτικών και των επαγγελματιών. Ένα αποτελεσματικό και προσβάσιμο εργαλείο διόρθωσης γραμματικής έχει τη δυνατότητα να αναβαθμίσει τη γραπτή επικοινωνία σε όλους αυτούς τους τομείς, δίνοντας στους χρήστες τη δυνατότητα να μεταφέρουν τις ιδέες τους με ακρίβεια και αντίκτυπο.

Εν κατακλείδι, αναδεικνύεται η ικανότητα ενός λεπτομερώς ρυθμισμένου μοντέλου T5 να ανταποκρίνεται στις προκλήσεις της διόρθωσης της γραμματικής. Οι γνώσεις που αποκτήθηκαν ανοίγουν το δρόμο για μελλοντική έρευνα όσον αφορά την τελειοποίηση των αρχιτεκτονικών των μοντέλων και την επέκταση του πεδίου εφαρμογής της αυτοματοποιημένης γλωσσικής διόρθωσης.

## 2. Background της GEC

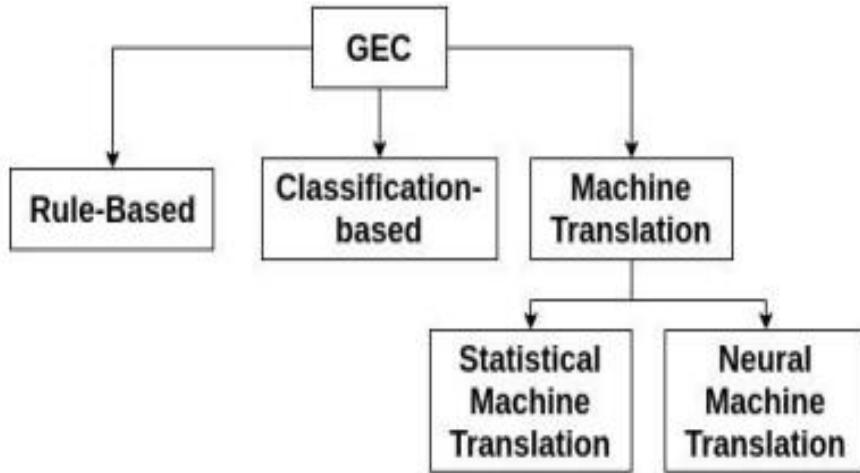
Η κατανόηση των ιδιαιτεροτήτων της γραμματικής και του συντακτικού παραμένει μια τρομερή πρόκληση, τόσο για τους μαθητές όσο και για τους φυσικούς ομιλητές της γλώσσας. Τα γραμματικά λάθη στο γραπτό κείμενο μπορούν να εμποδίσουν την κατανόηση, να μειώσουν τη συνολική ποιότητα του περιεχομένου και να εμποδίσουν ακόμη και την ακαδημαϊκή και επαγγελματική επιτυχία. Αναγνωρίζοντας την ευρεία παρουσία αυτών των λαθών και τον αντίκτυπό τους στην επικοινωνία, ο τομέας της διόρθωσης γραμματικών λαθών (GEC) έχει εξελιχθεί με την πάροδο του χρόνου. Εχουν διερευνηθεί πολυάριθμες προσεγγίσεις για τη διόρθωση της γραμματικής, που κυμαίνονται από συστήματα βασισμένα σε κανόνες (rule-based) έως στατιστικά μοντέλα (classification-based) και μηχανική μετάφραση (machine translation). Οι παραδοσιακές μέθοδοι, αν και χρήσιμες σε κάποιο βαθμό, συχνά δυσκολεύονται να αντιμετωπίσουν τις ιδιαιτερότητες και την εξαρτώμενη από τα συμφραζόμενα φύση των γραμματικών λαθών. Σε απάντηση αυτών, ο τομέας της Επεξεργασίας Φυσικής Γλώσσας (NLP) υπέστη μια αλλαγή με την έλευση μοντέλων βασισμένων σε νευρωνικά δίκτυα, προσφέροντας την υπόσχεση μιας ακριβέστερης και πιο συνειδητοποιημένης ως προς τα συμφραζόμενα γραμματικής διόρθωσης. Καθώς εμβαθύνουμε στις ιδιαιτερότητες της NLP, φωτίζουμε τα κίνητρα πίσω από τη μελέτη της και την επιτακτική ανάγκη για προηγμένες λύσεις με επίγνωση του πλαισίου που ανταποκρίνονται στις ποικίλες και εξελισσόμενες απαιτήσεις της σύγχρονης γλωσσικής χρήσης.



Εικόνα 2.1: Κατηγοριοποίηση των ερευνών σχετικά με τη GEC

## 2.1 Προσεγγίσεις της GEC

Η εμφάνιση των συστημάτων διόρθωσης γραμματικών σφαλμάτων (GEC) εγκαινίασε μια νέα εποχή, όπου η συγχώνευση της γλωσσολογικής ανάλυσης και της τεχνητής νοημοσύνης υπόσχεται να αυτοματοποιήσει την ανίχνευση και τη διόρθωση αυτών των γλωσσικών ατελειών. Για να κατανοήσει κανείς το ποικιλόμορφο τοπίο της GEC, πρέπει να εμβαθύνει στην εξέλιξη των προσεγγίσεων και των μεθοδολογιών της. Ξεκινώντας από τα στοιχειώδη συστήματα που βασίζονται σε κανόνες (rule-based), τα οποία έθεσαν τα θεμέλια για τον εντοπισμό και τη διόρθωση σφαλμάτων, προχωρούμε στη στατιστική εποχή (classification-based), όπου τα γλωσσικά μοντέλα προανήγγειλαν ένα σημαντικό άλμα προς τα εμπρός. Η ταξινόμηση (classification) και τα μοντέλα sequence-to-sequence έχουν επαναπροσδιορίσει τα πρότυπα εντοπισμού και διόρθωσης σφαλμάτων, καταλήγοντας στη χρήση των νευρωνικών δικτύων, όπου τα προ-εκπαιδευμένα γλωσσικά μοντέλα και οι αρχιτεκτονικές που βασίζονται στην προσοχή κυριαρχούν πλέον.



Εικόνα 2.2: Προσεγγίσεις της GEC

## Rule-based systems

Τα συστήματα που βασίζονται σε κανόνες (Rule-based systems) διασφαλίζουν ότι οι φράσεις αντιστοιχίζονται με συγκεκριμένα προκαθορισμένους και χειροκίνητα γραμμένους κανόνες γραμματικής (εσφαλμένα μοτίβα). Οι συγκεκριμένοι κανόνες βασίζονται συνήθως σε γραμματικές χωρίς συμφραζόμενα (Context Free Grammars), δηλαδή σε ένα σύνολο αναδρομικών κανόνων που χρησιμοποιούνται για τη δημιουργία μοτίβων συμβολοσειρών. Ωστόσο, περιλαμβάνεται και συντακτική ανάλυση (Syntactical analysis). Η χρήση της μας βοηθά να κατανοήσουμε το λογικό νόημα ορισμένων δεδομένων προτάσεων ή τμημάτων αυτών των προτάσεων. Στο σύστημα αυτό χρησιμοποιείται ένας αναλυτής (Parser) για να διαπιστωθεί αν το επισημειωμένο κείμενο στο κομμάτι της ομιλίας (Part of Speech Tags) ακολουθεί τους κανόνες [2]. Οι φράσεις θεωρούνται λανθασμένες εάν ταιριάζουν με τουλάχιστον έναν από τους κανόνες. Η μέθοδος αυτή είναι η απλούστερη στην εφαρμογή από τις τρεις, ωστόσο αν και έχει πολλά πλεονεκτήματα, παρουσιάζει και αρκετά μειονεκτήματα [3].

## Πλεονεκτήματα

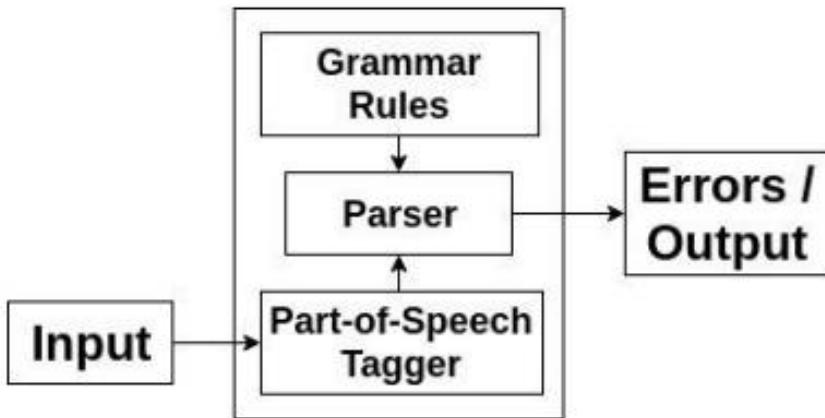
1. Απλή εφαρμογή.
2. Πολύ ακριβής και δεν απαιτεί μεγάλη ποσότητα δεδομένων εκπαίδευσης, γεγονός που αποτελεί σημαντικό πλεονέκτημα, ιδίως για γλώσσες με περιορισμένους πόρους. Αυτό την καθιστά σχετικά εύκολη να αναπτυχθεί για λιγότερο διαδεδομένες γλώσσες.

3. Το λογισμικό θα μπορούσε να ελέγξει το κείμενο και να δώσει άμεση ανατροφοδότηση ενώ πληκτρολογείται η πρόταση, ακόμη και πριν αυτή ολοκληρωθεί.
4. Εύκολη στη διαμόρφωση και μπορεί να ενεργοποιηθεί/απενεργοποιηθεί μεμονωμένα.
5. Μπορεί να προτείνει διεξοδικά και ευεργετικά μηνύματα σφάλματος. Όταν τα λάθη ακολουθούν συνεπείς κανόνες, τα συστήματα αυτά μπορούν να επιτύχουν υψηλό επίπεδο ακρίβειας.
6. Εύκολα κατανοητή και επεκτάσιμη. Τα συστήματα που βασίζονται σε κανόνες είναι ιδιαίτερα διαφανή. Κάθε διόρθωση βασίζεται σε έναν προκαθορισμένο κανόνα, καθιστώντας σαφές γιατί προτείνεται μια συγκεκριμένη διόρθωση. Αυτή η διαφάνεια είναι πολύτιμη για εκπαιδευτικούς σκοπούς, καθώς βοηθά τους χρήστες να κατανοήσουν και να μάθουν από τα λάθη τους.
7. Οι κανόνες μπορούν να προστίθενται σταδιακά.
8. Είναι η καλύτερη για απλές γλώσσες με ελάχιστους κανόνες γραμματικής.

## Μειονεκτήματα

1. Απαιτεί εκτεταμένη εργασία και τεχνογνωσία. Η ανάπτυξη rule-based συστημάτων GEC βασίζεται στην εμπειρογνωμοσύνη των γλωσσολόγων, των γλωσσικών επιστημόνων ή των σχεδιαστών κανόνων. Η δημιουργία και η διατήρηση ενός ολοκληρωμένου συνόλου κανόνων μπορεί να απαιτεί εντατική εργασία και γνώσεις ειδικού τομέα.
2. Καθώς η γραμματική γίνεται πολύπλοκη, οι κανόνες γίνονται πολύπλοκοι. Η πολυπλοκότητα της γραμματικής αυξάνεται καθώς σύνθετες περιπτώσεις σφαλμάτων εμφανίζονται. Τα συστήματα που βασίζονται σε κανόνες έχουν συνήθως ρηχή κατανόηση του πλαισίου. Μπορεί να δυσκολεύονται να αντιμετωπίσουν σφάλματα που απαιτούν κατανόηση του ευρύτερου πλαισίου ή της σημασιολογικής σημασίας. Τα σύνθετα σφάλματα που περιλαμβάνουν πολλές προτάσεις μπορεί να είναι δύσκολο να αντιμετωπιστούν.
3. Όλα ή πολλά λάθη δεν μπορούν να αντιμετωπιστούν με κανόνες. Τα rule-based συστήματα ενδέχεται να μην αποδίδουν καλά σε σπάνια ή ειδικά για το περιβάλλον μοτίβα σφαλμάτων που δεν συμμορφώνονται με προκαθορισμένους κανόνες. Τα συστήματα αυτά ενδέχεται να μην έχουν την ευελιξία να χειριστούν ασυνήθιστη γλωσσική χρήση, αφήνοντας ορισμένα μοτίβα σφαλμάτων χωρίς αντιμετώπιση.

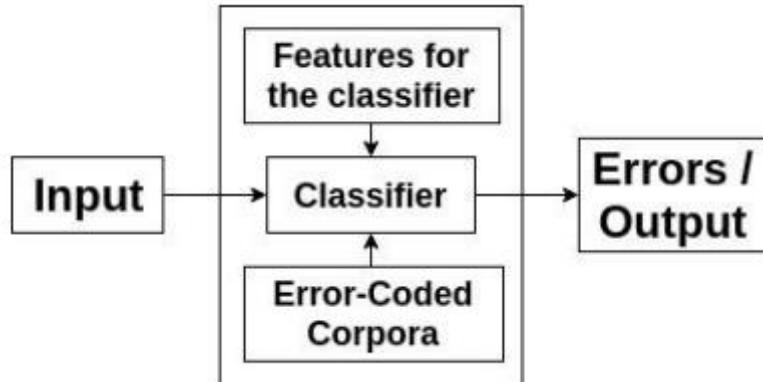
4. Απαιτεί λεπτομερείς κανόνες γραμματικής προκειμένου να χειριστεί όλους τους τύπους δομής προτάσεων.
5. Καθώς οι κανόνες αυξάνονται, γίνεται δύσκολη η συντήρηση και η ερμηνεία. Οι γλώσσες είναι συχνά διφορούμενες και τα συστήματα που βασίζονται σε κανόνες ενδέχεται να δυσκολεύονται να επιλύσουν αποτελεσματικά την ασάφεια. Μπορεί να παράγουν εσφαλμένες διορθώσεις όταν έρχονται αντιμέτωπα με πολλαπλές έγκυρες ερμηνείες. Επιπλέον, καθώς η γλώσσα εξελίσσεται, τα συστήματα αυτά απαιτούν συνεχείς ενημερώσεις και συντήρηση για να προσαρμόζονται στην μεταβαλλόμενη χρήση της γλώσσας και στα αναδυόμενα πρότυπα σφαλμάτων. Αυτό μπορεί να είναι απαιτητικό σε πόρους.



Εικόνα 2.3: Στοιχεία συστήματος βασισμένου σε κανόνες

## Classification-based systems

Η διαθεσιμότητα μεγάλου σώματος κωδικοποιημένων λαθών (error-coded corpora) επέτρεψε στους ερευνητές να χρησιμοποιήσουν περισσότερες data-driven προσεγγίσεις για το GEC. Χρησιμοποιήθηκαν αλγόριθμοι μηχανικής μάθησης με σκοπό τη δημιουργία ταξινομητών για τη διόρθωση συγκεκριμένων τύπων σφαλμάτων. Το μοντέλο μέγιστης εντροπίας (Maximum Entropy - ME) χρησιμοποιήθηκε για τον προσδιορισμό της καλύτερης δυνατής λέξης/αντικατάστασης σε συνδυασμό με τα αρχικά δείγματα, ενώ κατά καιρούς έχουν χρησιμοποιηθεί και άλλοι ταξινομητές όπως οι Naive Bayes, Decision Tree και Averaged Perceptron. Σε αυτή την προσέγγιση που βασίζεται σε ταξινομητή, οι πιθανοί υποψήφιοι δηλαδή οι λέξεις/αντικαταστάσεις αντιμετωπίζονται ως ετικέτες κλάσης, και τα περιβάλλοντα n-grams, οι ετικέτες PoS (Part of Speech) και οι γραμματικές σχέσεις χρησιμοποιούνται ως γνωρίσματα (features).



Εικόνα 2.4: Στοιχεία συστήματος βασισμένου σε ταξινομητή

Αυτοί οι ταξινομητές χρησιμοποιήθηκαν για τον εντοπισμό λαθών σε άρθρα και πέτυχαν ακρίβεια 88%. Με την πάροδο των χρόνων, χρησιμοποιήθηκαν για την ανίχνευση λαθών και σε άλλα πράγματα όπως λάθη σε μορφές ρημάτων, προθέματα, αριθμό ουσιαστικών, συμφωνία υπο-ρήματος, κόμμα και άλλα, καταλήγοντας να μπορούν να αντιμετωπίσουν σφάλματα κάθε είδους [4]. Επίσης, λόγω του ότι τα γνωρίσματα για τον ταξινομητή εξαρτώνται από τον τύπο σφάλματος, ένας ταξινομητής μπορεί να ανιχνεύσει μόνο έναν τύπο σφάλματος. Και αυτό υποθέτωντας ότι η υπόλοιπη πρόταση είναι απαλλαγμένη από σφάλματα και ότι το τρέχον σφάλμα είναι ανεξάρτητο, πράγμα που συνήθως δεν ισχύει. Η συνήθης προσέγγιση που χρησιμοποιείται για να ξεπεραστεί αυτό το πρόβλημα είναι η δημιουργία πολλαπλών ταξινομητών, καθένας από τους οποίους διορθώνει έναν τύπο σφάλματος και αυτή η συλλογή ταξινομητών στη συνέχεια συντίθεται σε έναν αγωγό.

Target Error Type	Features	Year
article	Linguistic Feature Engineering	2006
preposition		2007
		2008
		2010
		2008
		2008
article, preposition		2010
		2010
article, preposition, verb form, none number, sub-verb agreement		2013
article, preposition, verb form, none number, sub-verb agreement word form, orthography and punctuation, style		2014
article	Deep Learning	2015
article, preposition, verb form, none number, sub-verb agreement		2017
		2018
article, preposition, verb form, noun number, sub-verb agreement, comma		2019
all		2019

Εικόνα 2.5: Συστήματα GEC με βάση την ταξινόμηση

Όμως αυτή η προσέγγιση δεν λειτουργεί καλά στην περίπτωση εξαρτώμενων σφαλμάτων. Για την επίλυση του προβλήματος των εξαρτώμενων σφαλμάτων, οι ερευνητές συνδύασαν την προσέγγιση με βάση την ταξινόμηση και την προσέγγιση στατιστικής μηχανικής μετάφρασης (Statistical Machine Translation Approach), όπου θα συζητηθεί παρακάτω. Χρησιμοποιώντας αυτή τη μέθοδο, η αρχική πρόταση αποκωδικοποιείται σε πολλαπλές πιθανές διορθώσεις της (που ονομάζονται επίσης υποθέσεις) με σκοπό την εύρεση της καλύτερης δυνατής διόρθωσης για την πρόταση. Αυτή η διαδικασία αποκωδικοποίησης πραγματοποιείται σε επαναλήψεις και σε κάθε επανάληψη η πρόταση από την προηγούμενη επανάληψη βελτιώνεται κάνοντας σταδιακές αλλαγές. Οι προτάσεις διατηρούνται στην επόμενη επανάληψη με βάση τη βαθμολογία της τρέχουσας επανάληψης. Ο αποκωδικοποιητής χρησιμοποιεί τη γραμματική ορθότητα και ομαλότητα για να βαθμολογήσει τις προτάσεις. Η διαδικασία επαναλαμβάνεται έως ότου δεν απομείνουν προτάσεις ή έως ότου επιτευχθεί ο μέγιστος αριθμός επαναλήψεων.

Παρόλο όμως που οι προσεγγίσεις ταξινόμησης ήταν κάποτε δημοφιλείς, δεν υιοθετούνται συνήθως σήμερα λόγω των πολλών μειονεκτημάτων που παρουσιάζουν, γι' αυτό εξετάζουμε μερικές τυπικές εργασίες σε γενικές γραμμές.

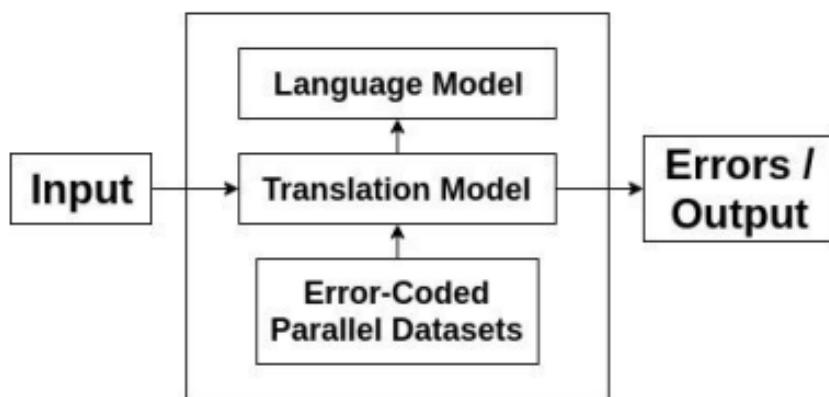
## Μειονεκτήματα

1. Δεδομένου ότι κάθε ταξινομητής διορθώνει έναν μόνο τύπο σφάλματος, αυτό αγνοεί τις εξαρτήσεις μεταξύ των λέξεων σε μια πρόταση. Εάν ένα μοντέλο ταξινόμησης ταξινομήσει εσφαλμένα μια λέξη ή φράση ως σωστή, αυτό μπορεί να οδηγήσει σε διάδοση σφάλματος, όπου το μοντέλο αποτυγχάνει να διορθώσει τα επόμενα σφάλματα που εξαρτώνται από την προηγούμενη διόρθωση.
2. Εξαρτώμενος από το περιβάλλον περιεχόμενο, ο ταξινομητής υποθέτει εμμέσως ότι το περιβάλλον περιεχόμενο είναι απαλλαγμένο από γραμματικά λάθη, κάτι που συχνά δεν συμβαίνει. Η δυσκολία να κατανοήσει το ευρύτερο πλαίσιο στο οποίο συμβαίνουν τα λάθη είναι ένα γεγονός που μπορεί να οδηγήσει σε λανθασμένες διορθώσεις.
3. Απαιτεί χειροκίνητα επεξεργασμένα γνωρίσματα, κάτι το οποίο είναι αρκετά χρονοβόρο και χρειάζεται γνώση εμπειρογνωμόνων.

## Machine translation

### Statistical Machine Translation

Πριν από την προσφυγή στη SMT (Statistical Machine Translation), οι διορθώσεις γραμματικών λαθών επιτυγχάνονταν κυρίως με μεθόδους που βασίζονται σε κανόνες ή σε ταξινόμηση, οι περιορισμοί των οποίων είναι προφανείς. Στην παρούσα ενότητα, παρουσιάζουμε μοντέλα βασισμένα στη SMT και την ανάπτυξή τους στη διόρθωση γραμματικών λαθών, παρόλο που οι περισσότεροι σημερινοί ερευνητές επικεντρώνονται σε μοντέλα που βασίζονται στη νευρωνική αυτόματη μετάφραση (neural machine translation).



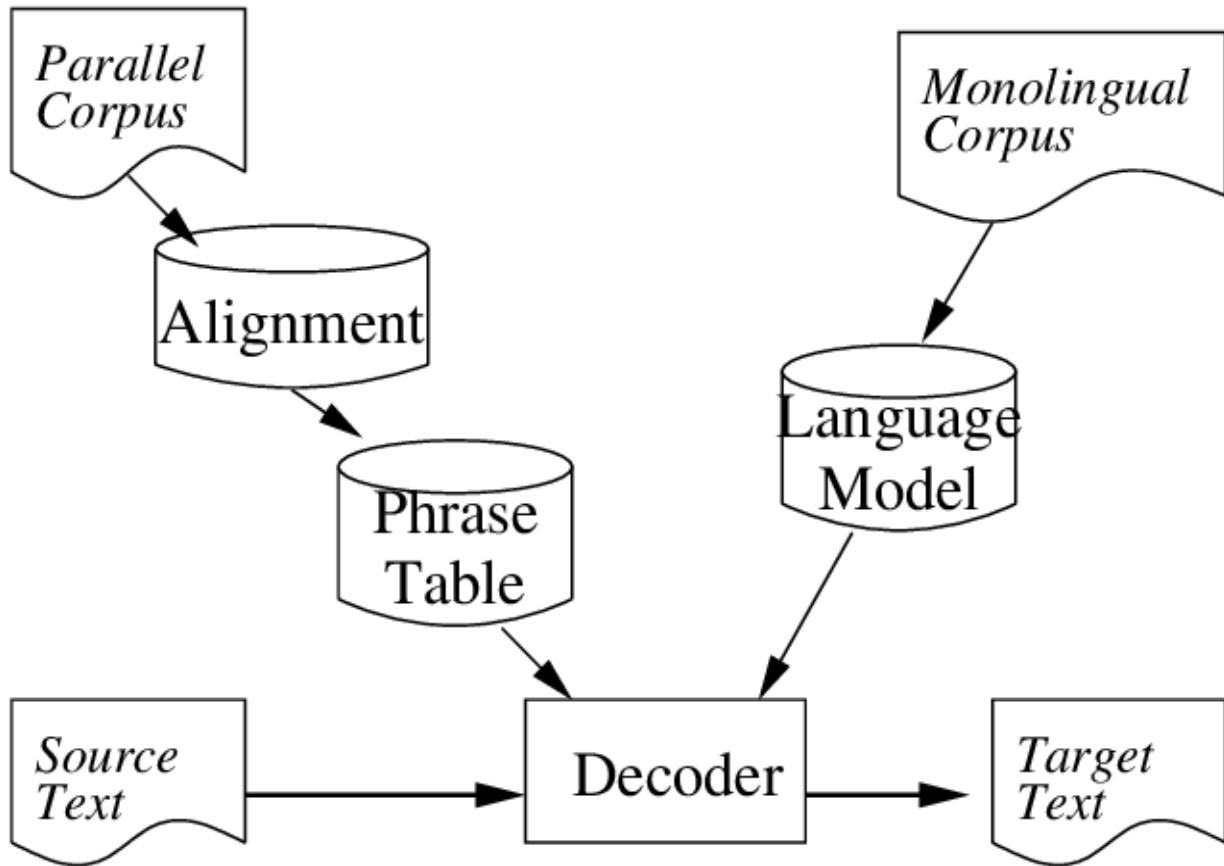
Εικόνα 2.6: Στοιχεία συστήματος στατιστικής μηχανικής μετάφρασης

Τα μοντέλα SMT χρησιμοποιήθηκαν για πρώτη φορά το 2006, διορθώνοντας ένα σύνολο 14 μετρήσιμων/μη μετρήσιμων σφαλμάτων. Χρησιμοποιήθηκε ένα μοντέλο SMT βασισμένο σε δένδρα εξαρτήσεων με τεχνητά κατασκευασμένο παράλληλο σώμα δεδομένων ως αποτέλεσμα της έλλειψης παράλληλων δεδομένων (error-coded parallel dataset) που απαιτούνται από την SMT. Το σύνολο δεδομένων που προέκυψε ονομάστηκε Chinese Learner Error Corpus (CLEC). Δοκιμάζοντας στο σύνολο δεδομένων CLEC, το μοντέλο πέτυχε ακρίβεια 61,52% στη διόρθωση λαθών και δείχνει την πολλά υποσχόμενη εφαρμογή της SMT, με τη βοήθεια παράλληλων δεδομένων GEC, για την επίλυση γενικότερων λαθών. Πριν από αυτό, η χρήση των SMT στην έρευνα της γραμματικής διόρθωσης ήταν σε μεγάλο βαθμό συγκρατημένη εξαιτίας της έλλειψης παράλληλων συνόλων δεδομένων με σχολιασμένα σφάλματα.

We took a	<b>guide</b>	tour	<b>on</b>		<b>center</b>	<b>city</b>
We took a	<b>guided</b>	tour	<b>of</b>	<b>the</b>	<b>city</b>	<b>center</b>

Εικόνα 2.7: Ένα δείγμα παράλληλων δεδομένων με μια αρχική λανθασμένη πρόταση (πάνω) και τη διόρθωσή της (κάτω)

Το 2012, τα μοντέλα SMT χρησιμοποιήθηκαν για την επίλυση όλων των τύπων σφαλμάτων, οι ερευνητές προσπάθησαν να χρησιμοποιήσουν ένα μεγάλης κλίμακας σύνολο εκμάθησης και προσπάθησαν να προσδιορίσουν την επίδραση του μεγέθους του συνόλου εκμάθησης στην προσέγγιση SMT με βάση τη φραστική γλώσσα (phrasal-based SMT approach). Τα αποτελέσματα έδειξαν ότι τα σφάλματα μπορούν να ταξινομηθούν σε δύο τύπους: Σφάλματα που μπορούν να επιλυθούν πιο αποτελεσματικά με την αύξηση του μεγέθους του σώματος, και ο δεύτερος τύπος σφαλμάτων εξαρτώταν από τις εκτεταμένες πληροφορίες πλαισίου. Το 2014, χρησιμοποιήθηκε μια υβριδική προσέγγιση η οποία ενσωμάτωσε ένα σύστημα βασισμένο σε κανόνες και ένα phrasal-based σύστημα SMT. Το σύστημα που βασίζεται σε κανόνες παράγει ένα σύνολο πιθανών υποψηφίων και στη συνέχεια χρησιμοποιείται γλωσσικό μοντέλο (language model) για να βρεθεί η πιθανότητα κάθε ενός από αυτούς και επιλέγει αυτόν με την υψηλότερη πιθανότητα. Αργότερα, χρησιμοποιήθηκαν και άλλες υβριδικές προσεγγίσεις με κυριότερη το συνδυασμό classification-based και statistical machine translation based συστημάτων.



Εικόνα 2.8: Στοιχεία Phrase-based SMT συστήματος

## Neural Machine Translation

Παρόλο που η προσέγγιση που βασίζεται στην SMT επωφελείται από την ικανότητά της να ενσωματώνει τον μεγάλο όγκο παράλληλων και μονόγλωσσων δεδομένων καθώς και τα βιοηθητικά στοιχεία νευρωνικών δικτύων, εξακολουθεί να υποφέρει από την έλλειψη πληροφοριών σχετικά με το πλαίσιο και την περιορισμένη ικανότητα γενίκευσης. Ως λύση, πολλοί ερευνητές αρχίζουν να ερευνούν προσεγγίσεις βασισμένες σε NMT για τη GEC. Με τις αυξανόμενες επιδόσεις που επιτυγχάνονται από τα μοντέλα νευρωνικού κωδικοποιητή-αποκωδικοποιητή (Encoder - Decoder) στη μηχανική μετάφραση, υιοθετούνται και τροποποιούνται μοντέλα βασισμένα σε νευρωνικό κωδικοποιητή-αποκωδικοποιητή. Σε σύγκριση με τα συστήματα GEC που βασίζονται σε SMT, τα μοντέλα που βασίζονται σε NMT έχουν δύο πλεονεκτήματα. Πρώτον, το μοντέλο νευρωνικού κωδικοποιητή-αποκωδικοποιητή μαθαίνει τις αντιστοιχίες από την πηγή (source) στο στόχο (target) απευθείας από τα παράλληλα δεδομένα εκπαίδευσης, και δεν χρειάζεται τα απαιτούμενα γνωρίσματα όπως στην SMT για να συλλάβει τις τακτικές της αντιστοιχίσης. Δεύτερον, τα συστήματα που βασίζονται σε NMT είναι σε θέση να διορθώνουν αφανείς ανορθόγραφες φράσεις και προτάσεις πιο αποτελεσματικά από τις προσεγγίσεις που βασίζονται σε SMT, αυξάνοντας την ικανότητα γενίκευσης. Όλα τα νευρωνικά συστήματα GEC βασίζονται στο μοντέλο κωδικοποιητή-αποκωδικοποιητή (ED), με μια εξαίρεση που βασίζεται στο μοντέλο παράλληλης επαναληπτικής επεξεργασίας (PIE).

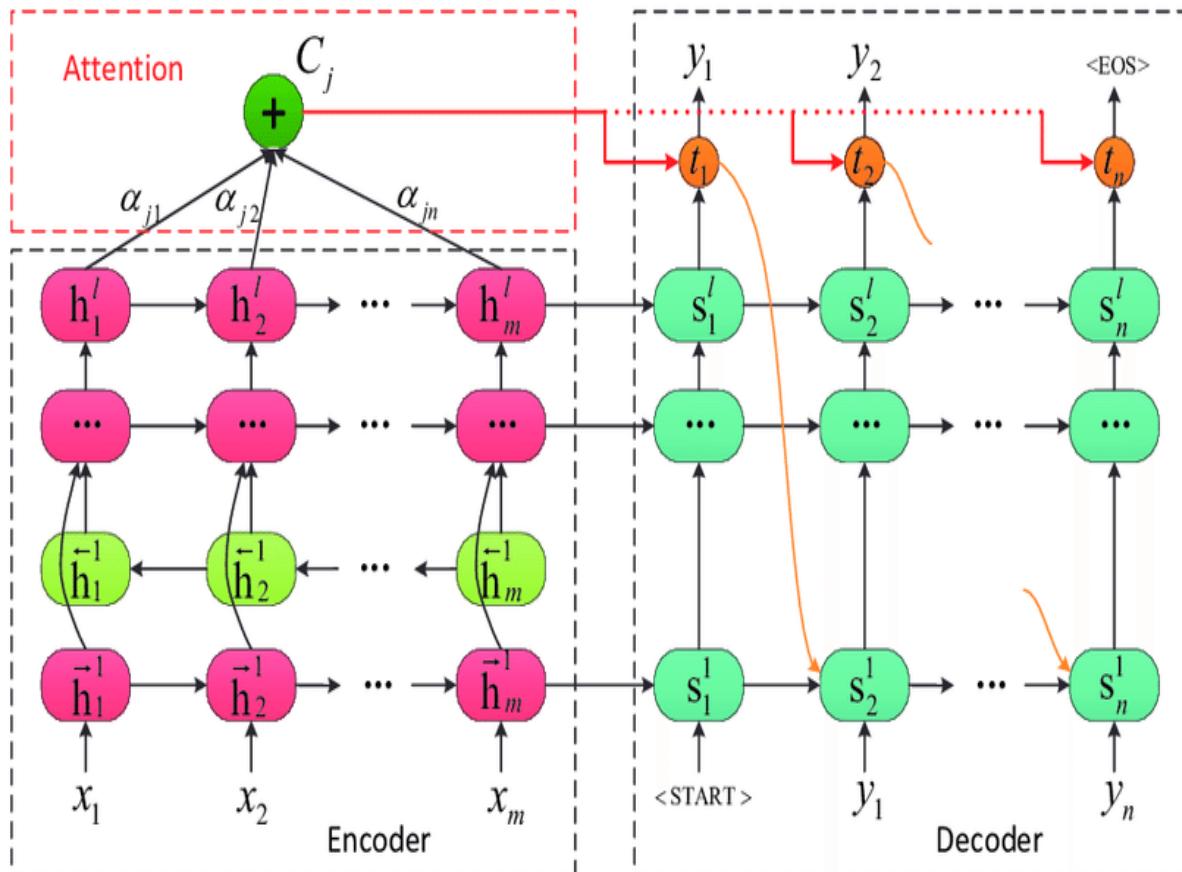
Model	Framework	Input	Handling Misspelling	Training LM	Year
ED	RNN	token level	alignment, word-level translation	-	2016
	RNN	character level	character-level translation model	Common Crawl	2016
	RNN	token level, character level	character-level translation model	Common Crawl	2017
	RNN	token level	spellchecker	-	2017
	CNN	token level	spellchecker	-	2018
	RNN	token level	spellchecker	Wikipedia	2018
	RNN	token level	spellchecker	Common Crawl	2018
	Transformer	token level	spellchecker	Common Crawl	2018
	Transformer	token level	spell error correction system	Common Crawl	2019
	CNN	token level, sentence level	spellchecker	-	2019
PIE	Transformer	token level	spellchecker	Wikipedia	2019
ED	Transformer	token level	-	-	2019

Εικόνα 2.9: Συστήματα GEC με βάση τη NMT. Η στήλη "Training LM" υποδεικνύει διαφορετικά μονόγλωσσα σώματα δεδομένων που χρησιμοποιήθηκαν για την εκπαίδευση των γλωσσικών μοντέλων

Η NMT για GEC προτάθηκε για πρώτη φορά το 2016 από τους Yuan και Briscoe. Τα συστήματα NMT όπως προαναφέρθηκε αναπτύσσονται χρησιμοποιώντας έναν μηχανισμό "κωδικοποιητή-αποκωδικοποιητή", όπου ένας κωδικοποιητής διαβάζει μια πρόταση και την κωδικοποιεί σε ένα διάνυσμα, και ο αποκωδικοποιητής εξάγει μια μετάφραση προβλέποντας την επόμενη λέξη με βάση το κωδικοποιημένο διάνυσμα και όλες τις προηγούμενες προβλεπόμενες λέξεις.

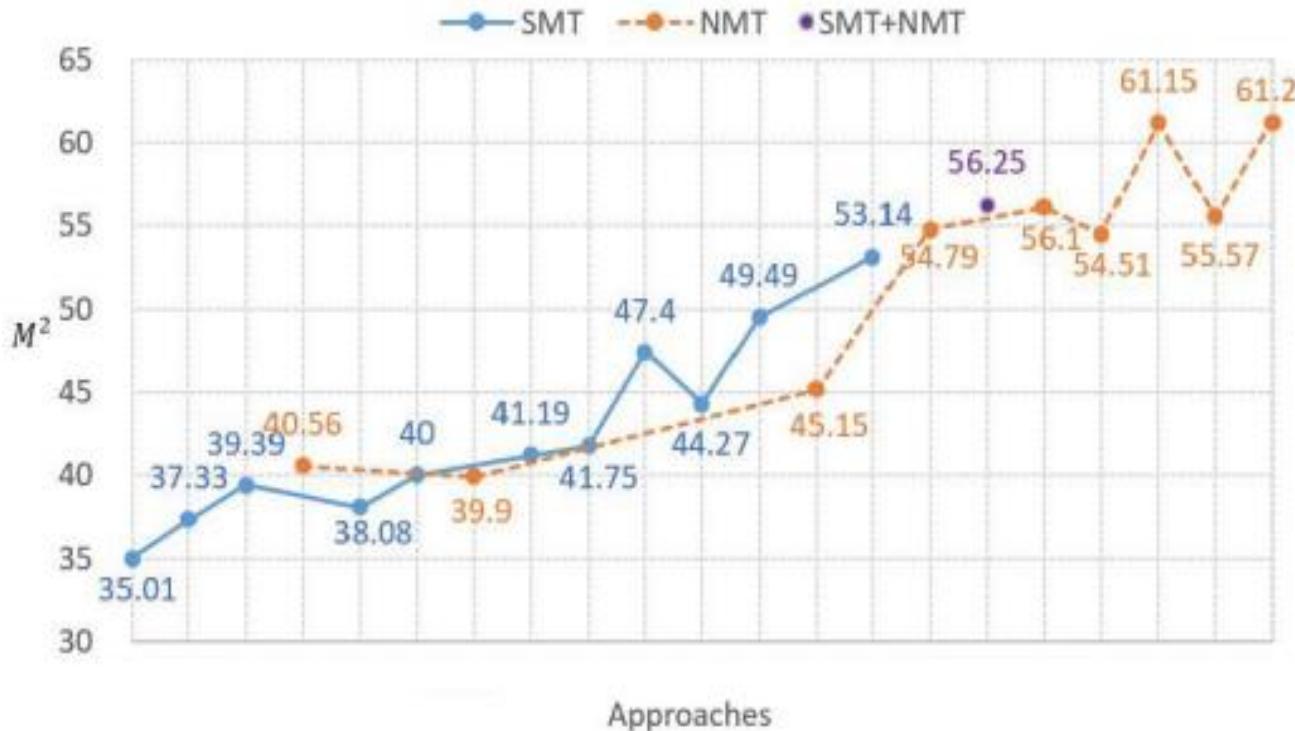
Οι Yuan και Briscoe χρησιμοποίησαν ένα αναδρομικό νευρωνικό δίκτυο (Recursive Neural Network), το οποίο περιείχε ένα "αμφίδρομο αναδρομικό νευρωνικό δίκτυο" ως

κωδικοποιητή και ένα μοντέλο "Attention-based" ως αποκωδικοποιητή. Το 2018, οι Chollampatt και Ng χρησιμοποίησαν ένα "πολυστρωματικό νευρωνικό δίκτυο συνελικτικού κωδικοποιητή-αποκωδικοποιητή", δηλώνοντας ότι τα CNN (convolutional neural networks) είναι πιο αποτελεσματικά στη σύλληψη του τοπικού πλαισίου από τα RNN, ότι τα πολλαπλά στρώματα συνέλιξης βοηθούν στη σύλληψη ευρύτερων πλαισίων και πως ο αριθμός των στρωμάτων συνέλιξης είναι 7 τόσο για τον κωδικοποιητή όσο και για τον αποκωδικοποιητή. Επίσης, το 2018, οι Roman και Marcin συνδύασαν τις προσεγγίσεις SMT και NMT για να επιτύχουν κορυφαία αποτελέσματα. Χρησιμοποίησαν ένα phrasal-based SMT σύστημα και ένα γλωσσικό μοντέλο 5-gram, ενώ για το σύστημα NMT χρησιμοποίησαν ένα μοντέλο κωδικοποιητή-αποκωδικοποιητή που βασίζεται στην προσοχή με έναν αμφίδρομο κωδικοποιητή και αποκωδικοποιητή ενός στρώματος. Η έξοδος που διορθώνεται από το σύστημα SMT περνά ως είσοδος στο μοντέλο NMT. Αυτή η διασύνδεση και των δύο μοντέλων βελτιώνει την απόδοση αυξάνοντας την ανάκληση.



Εικόνα 2.10: Η αρχιτεκτονική του μοντέλου νευρωνικής μηχανικής μετάφρασης.

Παρακάτω, αναλύουμε τις προσεγγίσεις που έχουν εφαρμοστεί πάνω στην εργασία της διόρθωσης λαθών, αναφέροντας τα σκορ που παρουσιάζει η κάθε προσέγγιση, με τη χρήση της μετρικής  $M^2$ , και καταδεικνύοντας την εξέλιξη που υπάρχει διαμέσω αυτών.



Εικόνα 2.11: Εξέλιξη προσεγγίσεων μηχανικής μετάφρασης (SMT based και NMT based approaches)

## 2.2 Διαμοιρασμένες Εργασίες

Τα διαμοιρασμένα έργα της GEC έχουν συμβάλει σημαντικά στην ανάπτυξη των ερευνών της. Οι συμμετέχουσες ομάδες ενθαρρύνονται να υποβάλλουν τα συστήματα διόρθωσης σφαλμάτων τους για υψηλότερες βαθμολογίες στις δοκιμές, κατά τη διάρκεια των οποίων έχουν σημειωθεί πολλές πρόοδοι, ιδίως στις τεχνικές. Ενώ οι διαμοιρασμένες εργασίες HOO-2011, HOO-2012 και CoNLL-2013 επικεντρώνονται σε διάφορους συγκεκριμένους τύπους σφαλμάτων, οι ομάδες που συμμετείχαν στις διαμοιρασμένες εργασίες CoNLL-2014 και BEA-2019 καλούνται να διορθώσουν όλους τους τύπους σφαλμάτων.

## CoNLL-2014

Στη διαμοιρασμένη εργασία CoNLL-2014 συμμετείχαν 13 ομάδες από όλο τον κόσμο για να αξιολογήσουν τα συστήματα διόρθωσης γραμματικών λαθών σε ένα κοινό σύνολο δοκιμών, χρησιμοποιώντας μια κοινή μετρική αξιολόγησης και έναν κοινό βαθμολογητή. Τα καλύτερα συστήματα στην διαμοιρασμένη εργασία επιτυγχάνουν βαθμολογία  $F_{0.5}$  37,33% όταν βαθμολογείται χωρίς εναλλακτικές απαντήσεις και 45,57% με εναλλακτικές απαντήσεις. Κάτι τέτοιο, υποδεικνύει ότι υπάρχουν ακόμη πολλά περιθώρια βελτίωσης της ακρίβειας των συστημάτων διόρθωσης γραμματικών λαθών. Τα σύνολα δεδομένων αξιολόγησης και ο βαθμολογητής που χρησιμοποιήθηκαν στη συγκεκριμένη εργασία χρησίμευσαν ως σημείο αναφοράς για μελλοντικές έρευνες σχετικά με τη διόρθωση γραμματικών λαθών.

ID	Candidate	CoNLL2014		
		RA-WER	BLEU	M <sup>2</sup>
a	Ungrammatical	7.51	87.79	83.99
b	ASR transcription	31.15	60.11	0.00
c	(b) + ITN	22.01	70.28	65.76
d	(b) + LM rescoring	30.13	61.82	17.63
e	(d) + ITN	<b>20.70</b>	<b>72.43</b>	<b>68.37</b>

Εικόνα 2.12: Βασικά αποτελέσματα στην CoNLL-2014 χρησιμοποιώντας διαφορετικές μετρικές [5]

Πιο αναλυτικά, στόχος της διαμοιρασμένης εργασίας CoNLL-2014 είναι η αξιολόγηση αλγορίθμων και συστημάτων για την αυτόματη ανίχνευση και διόρθωση γραμματικών λαθών που υπάρχουν σε δοκίμια που γράφονται από μαθητές που θέλουν να μάθουν ως δεύτερη γλώσσα την αγγλική [6]. Σε κάθε συμμετέχουσα ομάδα δίνονται δεδομένα εκπαίδευσης με χειροκίνητο σχολιασμό και διορθώσεις γραμματικών λαθών. Τα δεδομένα δοκιμής αποτελούνται από νέα, τυφλά δοκίμια ελέγχου. Προεπεξεργασμένα δοκίμια ελέγχου, τα οποία έχουν τμηματοποιηθεί σε προτάσεις και έχουν διαχωριστεί σε λέξεις, διατίθενται επίσης στις συμμετέχουσες ομάδες. Κάθε ομάδα πρέπει να υποβάλει την έξοδο του συστήματός της που αποτελείται από τα αυτόματα διορθωμένα δοκίμια, σε μορφή τμηματοποιημένων προτάσεων και συμβολισμών. Τα γραμματικά λάθη αποτελούνται από πολλούς διαφορετικούς τύπους, συμπεριλαμβανομένων των άρθρων ή προσδιορισμών, των εμπρόθετων, της μορφής ουσιαστικού, της μορφής ρήματος, της συμφωνίας υποκειμένου-ρήματος, των αντωνυμιών, της επιλογής λέξεων, της δομής της πρότασης, τη στίξη, τα κεφαλαία γράμματα, κ.λπ. Ωστόσο, οι περισσότερες προγενέστερες δημοσιευμένες έρευνες σχετικά με τη διόρθωση γραμματικών λαθών επικεντρώνονται μόνο σε ένα μικρό αριθμό συχνά εμφανιζόμενων τύπων λαθών, όπως τα λάθη άρθρων και προθέματος (Han et al., 2006- Gamon, 2010, Rozovskaya and Roth, 2010- Tetreault et al., 2010- Dahlmeier and Ng, 2011b). Τα λάθη άρθρων και προθέματος ήταν επίσης οι μόνοι τύποι λάθους που εμφανίστηκαν στην διαμοιρασμένη εργασία HOO- 2012. Ομοίως, παρόλο που όλοι οι τύποι σφαλμάτων περιλαμβάνονταν στην διαμοιρασμένη εργασία HOO-2011, σχεδόν όλες οι συμμετέχουσες ομάδες ασχολήθηκαν μόνο με τα λάθη άρθρων και προθέματος (εκτός από τα λάθη ορθογραφίας και στίξης). Στην κοινή εργασία CoNLL-2013, οι τύποι λαθών επεκτάθηκαν ώστε να περιλαμβάνουν πέντε τύπους λαθών, που περιλαμβάνουν άρθρο ή προσδιορισμό, προθέματα, αριθμό ουσιαστικών, μορφή ρήματος και συμφωνία υποκειμένου-ρήματος. Άλλοι τύποι σφαλμάτων όπως τα λάθη επιλογής λέξης (Dahlmeier and Ng, 2011a) δεν αντιμετωπίστηκαν.

Στην διαμοιρασμένη εργασία CoNLL-2014, θεωρήθηκε ότι η κοινότητα είναι πλέον έτοιμη να ασχοληθεί με όλους τους τύπους λαθών. Έτσι η εργασία αυτή εμπεριέχει 28 τύπους σφαλμάτων όπως φαίνεται παρακάτω. Δεδομένου ότι υπάρχουν 28 τύποι σφαλμάτων στη διαμοιρασμένη εργασία μας σε σύγκριση με δύο στη HOO-2012 και πέντε στη CoNLL-2013, υπάρχει μεγαλύτερη πιθανότητα να συναντήσουμε πολλαπλά, αλληλεπιδρώντα λάθη σε μια πρόταση στο δικό μας διαμοιρασμένο έργο. Αυτό αυξάνει την πολυπλοκότητα του καθήκοντος.

Type	Description	Example
Vt	Verb tense	Medical technology during that time [is → was] not advanced enough to cure him.
Vm	Verb modal	Although the problem [would → may] not be serious, people [would → might] still be afraid.
V0	Missing verb	However, there are also a great number of people [who → who are] against this technology.
Vform	Verb form	A study in 2010 [shown → showed] that patients recover faster when surrounded by family members.
SVA	Subject-verb agreement	The benefits of disclosing genetic risk information [outweighs → outweigh] the costs.
ArtOrDet	Article or determiner	It is obvious to see that [internet → the internet] saves people time and also connects people globally.
Nn	Noun number	A carrier may consider not having any [child → children] after getting married.
Npos	Noun possessive	Someone should tell the [carriers → carrier's] relatives about the genetic problem.
Pform	Pronoun form	A couple should run a few tests to see if [their → they] have any genetic diseases beforehand.
Pref	Pronoun reference	It is everyone's duty to ensure that [he or she → they] undergo regular health checks.
Prep	Preposition	This essay will [discuss about → discuss] whether a carrier should tell his relatives or not.
Wci	Wrong collocation/idiom	Early examination is [healthy → advisable] and will cast away unwanted doubts.
Wa	Acronyms	After [WOWII → World War II], the population of China decreased rapidly.
Wform	Word form	The sense of [guilty → guilt] can be more than expected.
Wtone	Tone (formal/informal)	[It's → It is] our family and relatives that bring us up.
Srun	Run-on sentences, comma splices	The issue is highly [debatable, a → debatable. A] genetic risk could come from either side of the family.
Smod	Dangling modifiers	[Undeniable, → It is undeniable that] it becomes addictive when we spend more time socializing virtually.
Spar	Parallelism	We must pay attention to this information and [assisting → assist] those who are at risk.
Sfrag	Sentence fragment	<b>However, from the ethical point of view.</b>
Ssub	Subordinate clause	This is an issue [needs → that needs] to be addressed.
WOinc	Incorrect word order	[Someone having what kind of disease → What kind of disease someone has] is a matter of their own privacy.
WOadv	Incorrect adjective/adverb order	In conclusion, [personally I → I personally] feel that it is important to tell one's family members.
Trans	Linking words/phrases	It is sometimes hard to find [out → out if] one has this disease.
Mec	Spelling, punctuation, capitalization, etc.	This knowledge [maybe relavant → may be relevant] to them.
Rloc-	Redundancy	It is up to the [patient's own choice → patient] to disclose information.
Cit	Citation	Poor citation practice.
Others	Other errors	An error that does not fit into any other category but can still be corrected.
Um	Unclear meaning	Genetic disease has a close relationship with the <b>born gene</b> . (i.e., no correction possible without further clarification.)

Εικόνα 2.13: Οι 28 τύποι σφαλμάτων στο διαμοιρασμένο έργο

Τα δεδομένα εκπαίδευσης που παρέχονται στην διαμοιρασμένη μας εργασία είναι το σώμα NUCLE, το σώμα μαθητών του NUS (National University of Singapore). Η έλλειψη ενός χειροκίνητα σχολιασμένου και διορθωμένου σώματος μαθητικών κειμένων αγγλικής γλώσσας έχει αποτελέσει εμπόδιο στην πρόοδο της διόρθωσης γραμματικών λαθών, καθώς εμποδίζει τις συγκριτικές αξιολογήσεις σε ένα κοινό σύνολο δεδομένων ελέγχου benchmark. Το NUCLE δημιουργήθηκε ακριβώς για να καλύψει αυτό το κενό. Πρόκειται για μια συλλογή από 1.414 δοκίμια που γράφτηκαν από φοιτητές του Εθνικού Πανεπιστημίου της Σιγκαπούρης (NUS) που δεν είναι φυσικοί ομιλητές της αγγλικής γλώσσας. Τα δοκίμια γράφτηκαν ως απάντηση σε κάποιες προτροπές και καλύπτουν ένα ευρύ φάσμα θεμάτων, όπως η ρύπανση του περιβάλλοντος, η υγειονομική περίθαλψη κ.λπ. Τα γραμματικά λάθη σε αυτά τα δοκίμια έχουν διορθωθεί με το χέρι από επαγγελματίες καθηγητές αγγλικών στο NUS. Για κάθε περίπτωση γραμματικού λάθους, οι μετατοπίσεις των χαρακτήρων αρχής και τέλους του λανθασμένου κειμένου επισημαίνονται, και παρέχεται ο τύπος του σφάλματος και η συμβολοσειρά διόρθωσης. Ο χειροκίνητος σχολιασμός πραγματοποιείται με τη χρήση γραφικής διεπαφής χρήστη που κατασκευάστηκε ειδικά για το σκοπό αυτό. Οι σχολιασμοί σφαλμάτων αποθηκεύονται ως σχολιασμοί stand-off, σε μορφή SGML.

```
<MISTAKE start_par="5" start_off="11" end_par="5" end_off="19">
<TYPE>Wform</TYPE>
<CORRECTION>absolutely</CORRECTION>
</MISTAKE>
```

Εικόνα 2.14: Παράδειγμα σχολιασμού σφάλματος σε μορφή SGML

Παρόμοια με το CoNLL-2013, 25 φοιτητές του NUS, οι οποίοι δεν είναι μητρικοί ομιλητές αγγλικών, προσλήφθηκαν για να γράψουν νέα δοκίμια που θα χρησιμοποιηθούν ως test data στη διαμοιρασμένη εργασία. Κάθε φοιτητής έγραψε δύο δοκίμια ως απάντηση στις δύο υποδείξεις που τους δόθηκαν, ένα δοκίμιο ανά υπόδειξη. Η πρώτη υπόδειξη χρησιμοποιήθηκε επίσης στα δεδομένα εκπαίδευσης NUCLE, αλλά η δεύτερη προτροπή είναι εντελώς νέα και δεν χρησιμοποιήθηκε προηγουμένως. Ως αποτέλεσμα, συγκεντρώθηκαν 50 νέα δοκίμια για testing. Τα στατιστικά στοιχεία των test data παρουσιάζονται παρακάτω.

	<b>Training data (NUCLE)</b>	<b>Test data</b>
# essays	1,397	50
# sentences	57,151	1,312
# word tokens	1,161,567	30,144

Εικόνα 2.15: Στατιστικά στοιχεία για training και test δεδομένα

Η αναφορά σφάλματος στα test data πραγματοποιήθηκε ανεξάρτητα από δύο μητρικούς ομιλητές της αγγλικής γλώσσας. Ένας από αυτούς είναι καθηγητής στο Κέντρο Επικοινωνίας Αγγλικής Γλώσσας του NUS και ο άλλος είναι ανεξάρτητος γλωσσολόγος με εκτεταμένη προηγούμενη εμπειρία στην αναφορά σφαλμάτων σε δοκίμια Άγγλων μαθητών. Η κατανομή των σφαλμάτων στα test data μεταξύ των τύπων σφάλματος παρουσιάζεται παρακάτω. Στη συνέχεια, τα test data προεπεξεργάστηκαν με τον ίδιο τρόπο με το σώμα NUCLE. Τα προεπεξεργασμένα test data διατέθηκαν στις συμμετέχουσες ομάδες. Παρόμοια με την CoNLL-2013, τα test data και οι σημειώσεις σφαλμάτων τους στη διαμοιρασμένη εργασία CoNLL-2014 είναι ελεύθερα διαθέσιμα μετά την εργασία.

Error type	Training data (NUCLE)	%	Test data (Annotator 1)	%	Test data (Annotator 2)	%
Vt	3,204	7.1%	133	5.5%	150	4.5%
Vm	431	1.0%	49	2.0%	37	1.1%
V0	414	0.9%	31	1.3%	37	1.1%
Vform	1,443	3.2%	132	5.5%	91	2.7%
SVA	1,524	3.4%	105	4.4%	154	4.6%
ArtOrDet	6,640	14.8%	332	13.9%	444	13.3%
Nn	3,768	8.4%	215	9.0%	228	6.8%
Npos	239	0.5%	19	0.8%	15	0.5%
Pform	186	0.4%	47	2.0%	18	0.5%
Pref	927	2.1%	96	4.0%	153	4.6%
Prep	2,413	5.4%	211	8.8%	390	11.7%
Wci	5,305	11.8%	340	14.2%	479	14.4%
Wa	50	0.1%	0	0.0%	1	0.0%
Wform	2,161	4.8%	77	3.2%	103	3.1%
Wtone	593	1.3%	9	0.4%	15	0.5%
Srun	873	1.9%	7	0.3%	26	0.8%
Smod	51	0.1%	0	0.0%	5	0.2%
Spar	519	1.2%	3	0.1%	24	0.7%
Sfrag	250	0.6%	13	0.5%	5	0.2%
Ssub	362	0.8%	68	2.8%	10	0.3%
WOinc	698	1.6%	22	0.9%	54	1.6%
WOadv	347	0.8%	12	0.5%	27	0.8%
Trans	1,377	3.1%	94	3.9%	79	2.4%
Mec	3,145	7.0%	231	9.6%	496	14.9%
Rloc-	4,703	10.5%	95	4.0%	199	6.0%
Cit	658	1.5%	0	0.0%	0	0.0%
Others	1,467	3.3%	44	1.8%	49	1.5%
Um	1,164	2.6%	12	0.5%	42	1.3%
All types	44,912	100.0%	2,397	100.0%	3,331	100.0%

Εικόνα 2.16: Κατανομή τύπων σφαλμάτων των training και test data. Η αναφορά σφάλματος στα test data πραγματοποιήθηκε ανεξάρτητα από δύο μητρικούς ομιλητές της αγγλικής γλώσσας

Έχει υιοθετηθεί η  $F_{0.5}$  ως μετρική αξιολόγησης στην εργασία CoNLL-2014 αντί για το πρότυπο  $F_1$  που χρησιμοποιείται στην CoNLL-2013. Η  $F_{0.5}$  δίνει διπλάσια έμφαση στην ακρίβεια από ότι στην ανάκληση, ενώ το  $F_1$  ζυγίζει εξίσου την ακρίβεια και την ανάκληση. Όταν ένας έλεγχος γραμματικής τίθεται σε πραγματική χρήση, είναι σημαντικό οι προτεινόμενες διορθώσεις του να είναι πολύ ακριβείς προκειμένου να κερδίσει την αποδοχή του χρήστη. Η παραμέληση του να προτείνεις διόρθωση δεν είναι τόσο κακή όσο το να προτείνεις λανθασμένη διόρθωση. Παρόμοια με την CoNLL-2013, χρησιμοποιούμε το MaxMatch ( $M^2$ ) scorer ως επίσημο σκορ στην CoNLL-2014. Το  $M^2$  scorer αναζητά αποτελεσματικά ένα σύνολο επεξεργασιών του συστήματος που ταιριάζουν στο μέγιστο με το σύνολο των χρυσών προδιαγραφών που καθορίζονται από έναν παρατηρητή. Όλες οι υποβληθείσες έξοδοι των συστημάτων αξιολογήθηκαν χρησιμοποιώντας τον βαθμολογητή  $M^2$ , με βάση τις σημειώσεις σφάλματος που παρήχθησαν από τους παρατηρητές. Η ανάκληση (Recall), η ακρίβεια (Precision) και η μέτρηση  $F_{0.5}$  όλων των ομάδων παρουσιάζονται παρακάτω όπου παρατηρούμε πως οι επιδόσεις των ομάδων ποικίλουν σημαντικά.

<b>Team ID</b>	<b>Precision</b>	<b>Recall</b>	<b><math>F_{0.5}</math></b>
CAMB	39.71	30.10	37.33
CUUI	41.78	24.88	36.79
AMU	41.62	21.40	35.01
POST	34.51	21.73	30.88
NTHU	35.08	18.85	29.92
RAC	33.14	14.99	26.68
UMC	31.27	14.46	25.37
PKU*	32.21	13.65	25.32
NARA	21.57	29.38	22.78
SJTU	30.11	5.10	15.19
UFC*	70.00	1.72	7.84
IPN*	11.28	2.85	7.09
IITB*	30.77	1.39	5.90

Εικόνα 2.17: Τα σκορ (σε %) των συστημάτων. Οι ομάδες οι οποίες υπέβαλλαν τις εξόδους των συστημάτων τους μετά το πέρας του deadline, έχουν αστερίσκο δίπλα στο όνομά τους

## BEA-2019

Όσον αφορά την διαμοιρασμένη εργασία BEA-2019, είναι αδιαμφισβήτητο ότι σημειώθηκε σημαντική πρόοδος από την τελευταία διαμοιρασμένη εργασία για τη διόρθωση γραμματικών λαθών πριν από πέντε χρόνια. Ο μετασχηματιστής με βάση τη νευρωνική μηχανική μετάφραση αποδείχθηκε αποτελεσματικός και οι ομάδες γενικά σημείωσαν σημαντικά υψηλότερη βαθμολογία στην BEA-2019 από ότι στην προηγούμενη διαμοιρασμένη εργασία (CoNLL-2014). Αυτό είναι ακόμα πιο σημαντικό καθώς στην εργασία επίσης εισήχθη ένα νέο σώμα κειμένων, το Cambridge English Write & Improve + LOCNESS corpus, το οποίο περιέχει ένα πολύ ευρύτερο φάσμα κειμένων σε διαφορετικά επίπεδα ικανοτήτων σε σχέση με τα προηγούμενα σώματα.

Συνολικά, τα πιο επιτυχημένα συστήματα, όπως θα δούμε παρακάτω, υποβλήθηκαν από την UEDIN-MS και την Kakao&Brain, τα οποία κατέλαβαν αντίστοιχα την πρώτη και τη δεύτερη θέση τόσο στο κομμάτι Restricted όσο και στο κομμάτι Low Resources. Επιπλέον, η UEDIN-MS σημείωσε μόλις 5 φορές χαμηλότερη  $F_{0.5}$  βαθμολογία στο Low Resource Track (64.24) από το Restricted Track (69.47), γεγονός που έδειχνε ότι είναι δυνατή η δημιουργία ενός ανταγωνιστικού σύστημα GEC χωρίς μεγάλες ποσότητες δεδομένων εκπαίδευσης με ανθρώπινο σχολιασμό.

Metric	Corpus	Sentence	
	Pearson $r$	Spearman $\rho$	Kendall $\tau$
ERRANT	0.64	0.626	0.623
$M^2$	0.623	0.687	0.617
GLEU	0.691	0.407	0.567
I-measure	-0.25	-0.385	0.564

Εικόνα 2.18: Συσχέτιση μεταξύ διαφόρων μετρικών αξιολόγησης και ανθρώπινων αξιολογήσεων

Πιο αναλυτικά, το Building Educational Applications (BEA) 2019 Shared Task on Grammatical Error Correction (GEC) συνεχίζει την παράδοση των προηγούμενων διαμοιρασμένων εργασιών Helping Our Own (HOO) και Conference on Natural Language Learning (CoNLL) και ήταν κίνητρο της ανάγκης να επανεκτιμηθεί ο τομέας μετά από μια πενταετή παύση [7]. Παρόλο που έχει σημειωθεί σημαντική πρόοδος από το τέλος της τελευταίας διαμοιρασμένης εργασίας CoNLL-2014, τα πρόσφατα συστήματα έχουν εκπαιδευτεί, προσαρμοστεί και δοκιμαστεί σε διαφορετικούς συνδυασμούς μετρικών και σώματος. Συνεπώς, ένας από τους κύριους στόχους του έργου BEA-2019 είναι να παρέχει για άλλη μια φορά μια πλατφόρμα όπου τα συστήματα μπορούν να επανεκτιμηθούν υπό πιο ελεγχόμενες συνθήκες.

Μια άλλη σημαντική συμβολή της διαμοιρασμένης εργασίας BEA-2019 είναι η εισαγωγή ενός νέου επισημειωμένου συνόλου δεδομένων, του Cambridge English Write & Improve (W&I) και του σώματος LOCNESS, το οποίο έχει σχεδιαστεί για να αντιπροσωπεύει ένα πολύ ευρύτερο φάσμα επιπέδων και ικανοτήτων της αγγλικής γλώσσας σε σύγκριση με το προηγούμενο σώμα. Αυτό είναι σημαντικό διότι τα συστήματα έχουν παραδοσιακά δοκιμαστεί μόνο στο σύνολο δοκιμών CoNLL-2014, το οποίο περιέχει μόνο 50 δοκίμια (1.312 προτάσεις) σε 2 διαφορετικά θέματα που γράφτηκαν από 25 προπτυχιακούς φοιτητές του πανεπιστημίου της Σιγκαπούρης. Αντίθετα, το δοκιμαστικό σετ W&I+LOCNESS περιέχει 350 δοκίμια (4.477 προτάσεις) για περίπου 50 θέματα γραμμένα από 334 συγγραφείς από όλο τον κόσμο (συμπεριλαμβανομένων φυσικών ομιλητών της αγγλικής γλώσσας). Αυτή η ποικιλομορφία θα ενθαρρύνει την ανάπτυξη συστημάτων που μπορούν να γενικεύουσαν καλύτερα σε μη γνωστά δεδομένα.

Μια άλλη διαφορά από τις προηγούμενες εργασίες είναι η εισαγωγή των tracks, δηλαδή των Restricted, Unrestricted και Low Source track. Αν και τα επισημειωμένα δεδομένα ήταν σχετικά σπάνια πριν από πέντε χρόνια, από τότε έχουν γίνει πιο διαθέσιμα, οπότε μπορούμε τώρα να ελέγξουμε τους πόρους στους οποίους έχουν πρόσβαση οι συμμετέχοντες. Το Restricted κομμάτι είναι πιο κοντά στις αρχικές διαμοιρασμένες εργασίες, διότι καθορίζουμε με ακρίβεια ποια επισημειωμένα σύνολα δεδομένων από μαθητές θα πρέπει να χρησιμοποιούν οι συμμετέχοντες, ενώ το Unrestricted κομμάτι επιτρέπει τη χρήση οποιωνδήποτε και όλων των διαθέσιμων συνόλων δεδομένων. Το Low Source track, αντίθετα, περιορίζει σημαντικά την ποσότητα των επισημειωμένων δεδομένων που είναι διαθέσιμα στους συμμετέχοντες και ενθαρρύνει την ανάπτυξη συστημάτων που δεν βασίζονται σε μεγάλες ποσότητες human annotated (επισημειωμένων από ανθρώπους) σημειώσεων. Ένας από τους στόχους του Low Resource track είναι, επομένως, να διευκολύνει την έρευνα στη GEC για τις γλώσσες όπου δεν υπάρχουν επισημειωμένα σύνολα δεδομένων εκπαίδευσης.

Αυτή η διαμοιρασμένη εργασία εισάγει νέα σύνολα δεδομένων: το Cambridge English Write & Improve (W&I) και το σώμα LOCNESS. Το Write & Improve είναι μια διαδικτυακή πλατφόρμα που βοηθά τους μη ιθαγενείς μαθητές της αγγλικής γλώσσας στο γραπτό τους λόγο. Συγκεκριμένα, οι μαθητές από όλο τον κόσμο υποβάλλουν επιστολές, ιστορίες, άρθρα και δοκίμια σε απάντηση σε διάφορες προτάσεις, και το σύστημα W&I παρέχει αυτοματοποιημένη ανατροφοδότηση. Από το 2014, οι σχολιαστές W&I έχουν χειροκίνητα επισημειώσει μερικές από αυτές τις υποβολές με

διορθώσεις και επίπεδα ικανότητας. Αν και οι χρήστες μπορούν να υποβάλουν οποιοδήποτε είδος κειμένου στο σύστημα Write & Improve, τα κείμενα πρώτα φιλτράρονται πριν αποσταλούν στους παρατηρητές για να αφαιρεθούν, για παράδειγμα, αποσπάσματα δοκιμών, τεχνικά δοκίμια, αντιγραμμένο κείμενο ιστοσελίδας, κενά ή μικρά κείμενα και μη αγγλικό κείμενο.

Δεδομένου ότι η πλειοψηφία των ερευνών της GEC έχει παραδοσιακά επικεντρωθεί σε μη γηγενή σφάλματα, οι ερευνητές επίσης ήθελαν να ενσωματώσουν μερικά εγγενή λάθη στην εργασία. Για να το κάνουν αυτό, χρησιμοποίησαν το σώμα LOCNESS, μια συλλογή από περίπου 400 δοκίμια που γράφτηκαν από ιθαγενείς Βρετανούς και Αμερικανούς προπτυχιακούς φοιτητές πάνω σε διάφορα θέματα. Δεδομένου ότι αυτά τα δοκίμια ήταν συνήθως πολύ μεγαλύτερα από τα κείμενα που υποβλήθηκαν στο Write & Improve, αρχικά τα φίλτραραν για να αφαιρέσουν δοκίμια μεγαλύτερα από 550 λέξεις. Επίσης, αφαιρέσαν δοκίμια που περιείχαν ετικέτες XML με πρόβλημα μεταγραφής, όπως <quotation> και <illegible>.

Τελικά, επιλέχθηκαν 3.600 επισημειωμένες υποβολές από την W&I, τις οποίες διανέμουμε σε εκπαιδευτικά (training), αναπτυξιακά (dev) και δοκιμαστικά (test) σύνολα, όπως αντίστοιχα γίνεται και με το σώμα δεδομένων LOCNESS. Επιπλέον, επισημειώθηκαν τα δοκιμαστικά σύνολα 5 φορές για να λάβουμε καλύτερα υπόψη τις εναλλακτικές διορθώσεις.

	<b>A</b>	<b>B</b>	<b>C</b>	<b>N</b>	<b>Total</b>
<b>Train</b>					
Texts	1,300	1,000	700	-	3,000
Sentences	10,493	13,032	10,783	-	34,308
Tokens	183,684	238,112	206,924	-	628,720
<b>Dev</b>					
Texts	130	100	70	50	350
Sentences	1,037	1,290	1,069	998	4,384
Tokens	18,691	23,725	21,440	23,117	86,973
<b>Test</b>					
Texts	130	100	70	50	350
Sentences	1,107	1,330	1,010	1,030	4,477
Tokens	18,905	23,667	19,953	23,143	85,668
<b>Total</b>					
Texts	1,560	1,200	840	100	3,700
Sentences	12,637	15,652	12,862	2,018	43,169
Tokens	221,280	285,504	248,317	46,260	801,361

Εικόνα 2.19: Στατιστικά στοιχεία σώματος W&I (A, B, C) και LOCNESS (N)

Στο Restricted κομμάτι της εργασίας επιτρέπεται η χρήση πολλών υφιστάμενων σωμάτων δεδομένων, όπως τα NUCLE, Lang-8 και FCE. Δεδομένου ότι αυτά τα σώματα ήταν προηγουμένως διαθέσιμα μόνο σε διαφορετικές μορφές, χρειάζονται νέες τυποποιημένες εκδόσεις διαθέσιμες για την εργασία. Η μορφή που πρέπει να έχουν τα δεδομένα είναι η μορφή M2. Στην μορφή M2, μια γραμμή που ξεκινάει από το S υποδηλώνει μια αρχική πρόταση, ενώ μια γραμμή που ξεκινάει από το A υποδεικνύει μια αναφορά επεξεργασίας.

```

S This are a sentence .
A 1 2 || | R:VERB:SVA || | is || |-REQUIRED- || | NONE || | 0
A 3 3 || | M:ADJ || | good || |-REQUIRED- || | NONE || | 0
A 1 2 || | R:VERB:SVA || | is || |-REQUIRED- || | NONE || | 1
A -1 -1 || | noop || |-NONE- || | REQUIRED || | -NONE- || | 2

```

*Εικόνα 2.20: Παράδειγμα μορφής M2 με πολλαπλές σημειώσεις*

Δεδομένου ότι τα σώματα FCE και NUCLE επισημειώθηκαν σύμφωνα με διαφορετικά πλαίσια τύπου σφάλματος και τα σώματα Lang8 και W&I+LOCNESS δεν επισημειώθηκαν με τους τύπους σφαλμάτων καθόλου, επανασημειώθηκαν όλα τα σώματα αυτόματα χρησιμοποιώντας ERRANT. Το σκορ ERRANT (λειτουργεί σαν το σκορ M<sup>2</sup> σε σχέση με την κύρια μετρική αξιολόγησης F<sub>0.5</sub>) χρησιμοποιείται αντί του σκορ M<sup>2</sup>, επειδή το σκορ ERRANT μπορεί να παρέχει πολύ πιο λεπτομερή ανατροφοδότηση, π.χ. όσον αφορά την απόδοση σε συγκεκριμένους τύπους σφαλμάτων. Πιο συγκεκριμένα έγιναν τα παρακάτω:

1. Έγιναν tokenized τα σώματα FCE και W&I+LOCNESS χρησιμοποιώντας spaCy v1.9.0. Τα σώματα Lang-8 και NUCLE ήταν pro-tokenized.
2. Χρησιμοποιήθηκε ERRANT για την αυτόματη ταξινόμηση των ανθρώπινων επεξεργασιών σε παράλληλες προτάσεις στα σώματα FCE, NUCLE και W&I+LOCNESS.
3. Χρησιμοποιήθηκε ERRANT για την αυτόματη εξαγωγή και ταξινόμηση των επεξεργασιών σε παράλληλες Lang-8 προτάσεις.

Σημειώνεται ότι καθώς το Lang-8 δεν είναι επισημειωμένο με ρητές επεξεργασίες, αποτελείται μόνο από παράλληλα ζεύγη προτάσεων. Κατά συνέπεια, χρησιμοποιήθηκε το ERRANT για να ευθυγραμμίσουμε τις προτάσεις και να εξηγάγουμε τις επεξεργασίες αυτόματα.

**W&I+LOCNESS**

	<b>FCE (all)</b>	<b>Lang-8</b>	<b>NUCLE</b>	<b>Train</b>	<b>Dev</b>	<b>Test</b>
<b>Type</b>	<b>%</b>	<b>%</b>	<b>%</b>	<b>%</b>	<b>%</b>	<b>%</b>
M	21.00	26.41	19.09	25.29	26.32	24.86
R	64.39	59.99	59.04	61.43	61.23	63.40
U	11.47	13.60	19.31	10.69	10.21	10.34
UNK	3.13	0.00	2.57	2.59	2.24	1.41
ADJ	1.36	1.25	1.58	1.52	1.48	1.05
ADJ:FORM	0.28	0.19	0.27	0.24	0.21	0.18
ADV	1.94	3.37	1.95	1.51	1.51	1.45
CONJ	0.67	0.98	0.71	0.51	0.58	0.75
CONTR	0.32	0.99	0.11	0.30	0.39	0.32
DET	10.86	11.93	15.98	11.25	10.43	10.41
MORPH	1.90	1.62	3.14	1.85	2.07	2.50
NOUN	4.57	4.51	3.80	4.36	4.30	2.89
NOUN:INFL	0.50	0.18	0.12	0.12	0.13	0.28
NOUN:NUM	3.34	4.28	8.13	4.05	3.29	4.07
NOUN:POSS	0.51	0.35	0.61	0.60	0.87	0.93
ORTH	2.94	3.99	1.62	4.77	4.61	8.03
OTHER	13.26	26.62	25.65	12.76	12.84	15.69
PART	0.29	0.50	0.46	0.84	0.79	0.49
PREP	11.21	8.00	7.69	9.79	9.70	8.33
PRON	3.51	2.72	1.26	2.64	2.33	2.45
PUNCT	9.71	6.06	5.16	17.16	19.37	16.73
SPELL	9.59	4.45	0.26	3.74	5.07	4.63
UNK	3.13	0.00	2.57	2.59	2.24	1.41
VERB	7.01	6.52	4.31	5.86	5.27	5.09
VERB:FORM	3.55	2.56	3.49	3.56	3.09	3.10
VERB:INFL	0.19	0.15	0.01	0.04	0.07	0.12
VERB:SVA	1.52	1.58	3.47	2.23	1.94	2.28
VERB:TENSE	6.04	6.03	7.01	6.07	6.20	5.43
WO	1.82	1.18	0.66	1.64	1.25	1.40
Total Edits	52,671	1,400,902	44,482	63,683	7,632	-

Εικόνα 2.21: Οι ERRANT κατανομές τύπων σφαλμάτων των σωμάτων FCE, Lang-8, NUCLE και W&I+LOCNESS. Η κατανομή των στοιχείων δοκιμής W&I+LOCNESS είναι μέση και για τους 5 παραπορητές.

**Restricted**

<b>Group</b>	<b>Rank</b>	<b>Teams</b>	<b>TP</b>	<b>FP</b>	<b>FN</b>	<b>P</b>	<b>R</b>	<b>F<sub>0.5</sub></b>
1	1	UEDIN-MS	3127	1199	<b>2074</b>	72.28	60.12	<b>69.47</b>
	2	Kakao&Brain	2709	894	2510	<b>75.19</b>	51.91	69.00
2	3	LAIX	2618	960	2671	73.17	49.50	66.78
	4	CAMB-CLED	2924	1224	2386	70.49	55.07	66.75
	5	Shuyao	2926	1244	2357	70.17	55.39	66.61
3	6	YDGEC	2815	1205	2487	70.02	53.09	65.83
	7	ML@IITB	<b>3678</b>	1920	2340	<b>65.70</b>	<b>61.12</b>	64.73
	8	CAMB-CUED	2929	1459	2502	66.75	53.93	63.72
4	9	AIP-Tohoku	1972	902	2705	68.62	42.16	60.97
	10	UFAL	1941	942	2867	67.33	40.37	59.39
	11	CVTE-NLP	1739	811	2744	68.20	38.79	59.22
5	12	BLCU	2554	1646	2432	60.81	51.22	58.62
6	13	IBM	1819	1044	3047	63.53	37.38	55.74
7	14	TMU	2720	2325	2546	53.91	51.65	53.45
	15	qiuwenbo	1428	854	2968	62.58	32.48	52.80
8	16	NLG-NTU	1833	1873	2939	49.46	38.41	46.77
	17	CAI	2002	2168	2759	48.01	42.05	46.69
	18	PKU	1401	1265	2955	52.55	32.16	46.64
9	19	SolomonLab	1760	2161	2678	44.89	39.66	43.73
10	20	Buffalo	604	<b>350</b>	3311	63.31	15.43	39.06
11	21	Ramaiah	829	7656	3516	9.77	19.08	10.83

**Unrestricted**

<b>Group</b>	<b>Rank</b>	<b>Teams</b>	<b>TP</b>	<b>FP</b>	<b>FN</b>	<b>P</b>	<b>R</b>	<b>F<sub>0.5</sub></b>
1	1	LAIX	2618	960	2671	<b>73.17</b>	49.50	<b>66.78</b>
	2	AIP-Tohoku	2589	1078	2484	70.60	51.03	65.57
2	3	UFAL	2812	1313	2469	68.17	53.25	64.55
3	4	BLCU	<b>3051</b>	2007	<b>2357</b>	60.32	<b>56.42</b>	59.50
4	5	Aparecium	1585	1077	2787	59.54	36.25	52.76
5	6	Buffalo	699	<b>374</b>	3265	65.14	17.63	42.33
6	7	Ramaiah	1161	8062	3480	12.59	25.02	13.98

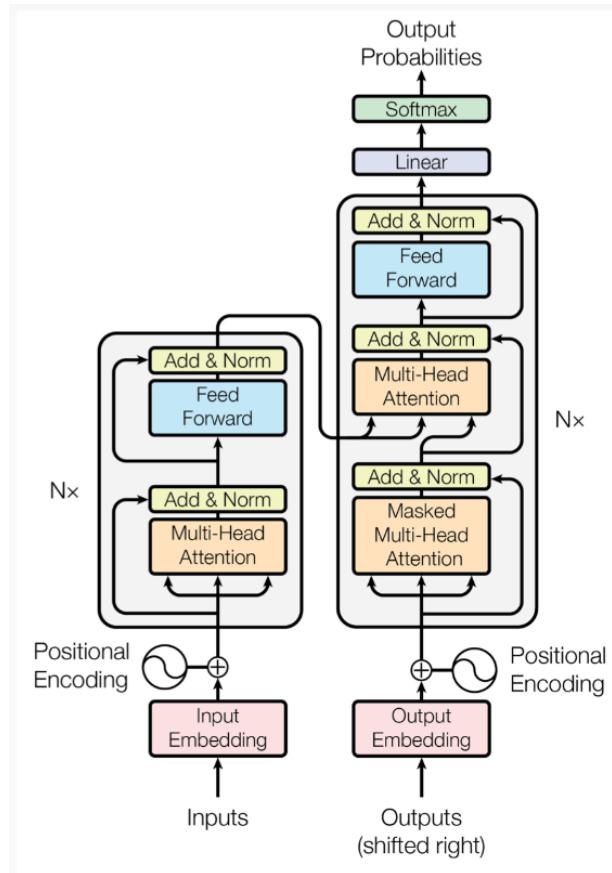
**Low Resource**

<b>Group</b>	<b>Rank</b>	<b>Teams</b>	<b>TP</b>	<b>FP</b>	<b>FN</b>	<b>P</b>	<b>R</b>	<b>F<sub>0.5</sub></b>
1	1	UEDIN-MS	2312	982	<b>2506</b>	<b>70.19</b>	<b>47.99</b>	<b>64.24</b>
2	2	Kakao&Brain	<b>2412</b>	1413	2797	63.06	46.30	58.80
3	3	LAIX	1443	<b>884</b>	3175	62.01	31.25	51.81
	4	CAMB-CUED	1814	1450	2956	55.58	38.03	50.88
4	5	UFAL	1245	1222	2993	50.47	29.38	44.13
5	6	Siteimprove	1299	1619	3199	44.52	28.88	40.17
	7	WebSpellChecker	2363	3719	3031	38.85	43.81	39.75
6	8	TMU	<b>1638</b>	4314	3486	27.52	31.97	28.31
7	9	Buffalo	446	1243	3556	26.41	11.14	20.73

Εικόνα 2.22: Τα επίσημα αποτελέσματα της BEA-2019 για όλες τις ομάδες σε όλα τα κομμάτια χρησιμοποιώντας την κύρια γενική διόρθωση ERRANT F<sub>0.5</sub>. Οι υψηλότερες τιμές (η χαμηλότερη για False Positive και False Negative) εμφανίζονται με έντονη γραμματοσειρά

## 2.3 Σχετικές Εργασίες

Προηγούμενες μελέτες έχουν δείξει ότι η GEC μπορεί να προσεγγιστεί ως μηχανική μετάφραση με τη χρήση ενός μοντέλου seq2seq (Luong et al., 2015 [8]) μαζί με αρχιτεκτονική Transformer (Vaswani et al., 2017 [9]) [10]. Ο Vaswani (2017) πρότεινε για πρώτη φορά το δίκτυο μετασχηματιστών για την επίλυση προβλημάτων μηχανικής μάθησης όπου οι είσοδοι είναι ακολουθίες και οι έξοδοι μπορεί να είναι ακολουθίες ή μια ετικέτα κλάσης (π.χ. μηχανική μετάφραση, ταξινόμηση συναισθημάτων). Ένα δίκτυο μετασχηματιστή αποτελείται από αρχιτεκτονικές κωδικοποιητή-αποκωδικοποιητή μαζί με μηχανισμούς προσοχής, χωρίς να χρησιμοποιεί καμία αναδρομική ή συνελικτική αρχιτεκτονική. Τα παραδοσιακά μοντέλα βαθιάς μάθησης για την επίλυση προβλημάτων μηχανικής μάθησης που σχετίζονται με ακολουθίες, απαιτούν την τροφοδοσία της εισόδου διαδοχικά. Στο δίκτυο μετασχηματιστή, όλες οι είσοδοι μπορούν να τροφοδοτούνται ταυτόχρονα. Η αρχιτεκτονική βασίζεται εξ ολοκλήρου σε έναν μηχανισμό προσοχής για να σχεδιάζει σφαιρικές εξαρτήσεις μεταξύ εισόδου και εξόδου. Αυτό επιτρέπει μεγαλύτερο παραλληλισμό στην εκπαίδευση, ταχύτερη εκμάθηση με αυξημένο όγκο δεδομένων και μικρότερη επίδραση της αύξησης του μήκους της ακολουθίας σε σύγκριση με τα παραδοσιακά μοντέλα βαθιάς μάθησης [11]. Εργασίες που βασίστηκαν πάνω στην αρχιτεκτονική Transformer για τη διόρθωση γραμματικών λαθών αναλύονται παρακάτω.



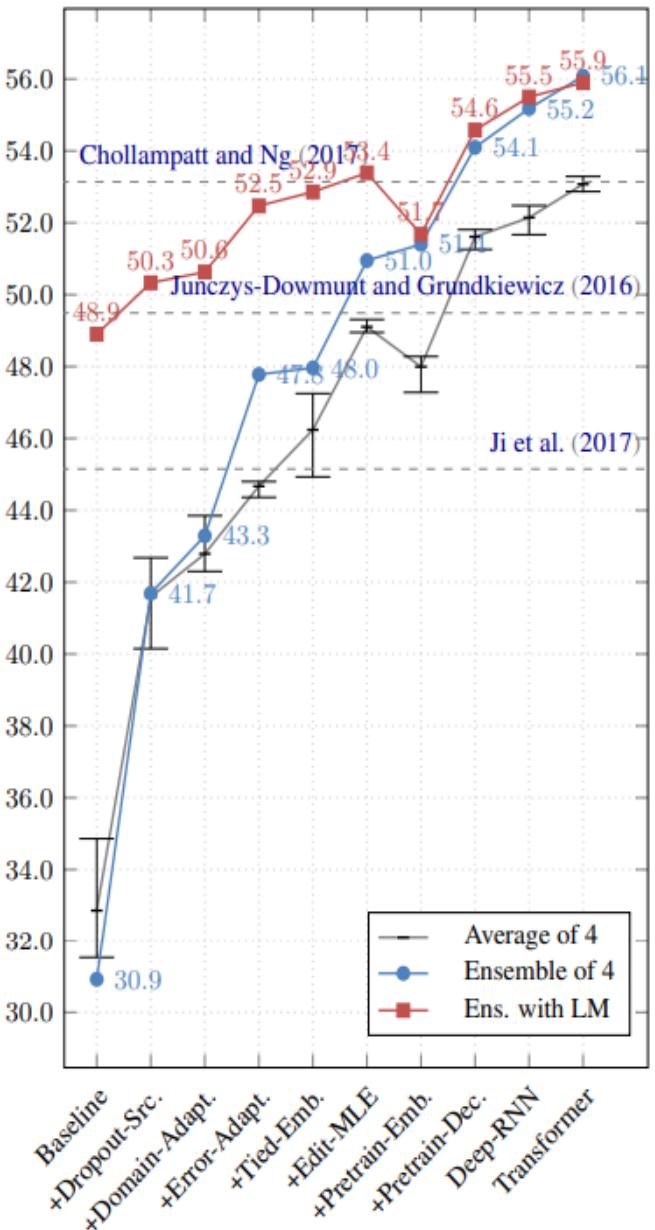
Εικόνα 2.23: Η αρχιτεκτονική του δικτύου μετασχηματιστών Transformer (Vaswani et al. 2017)

## Junczys-Dowmunt (2018)

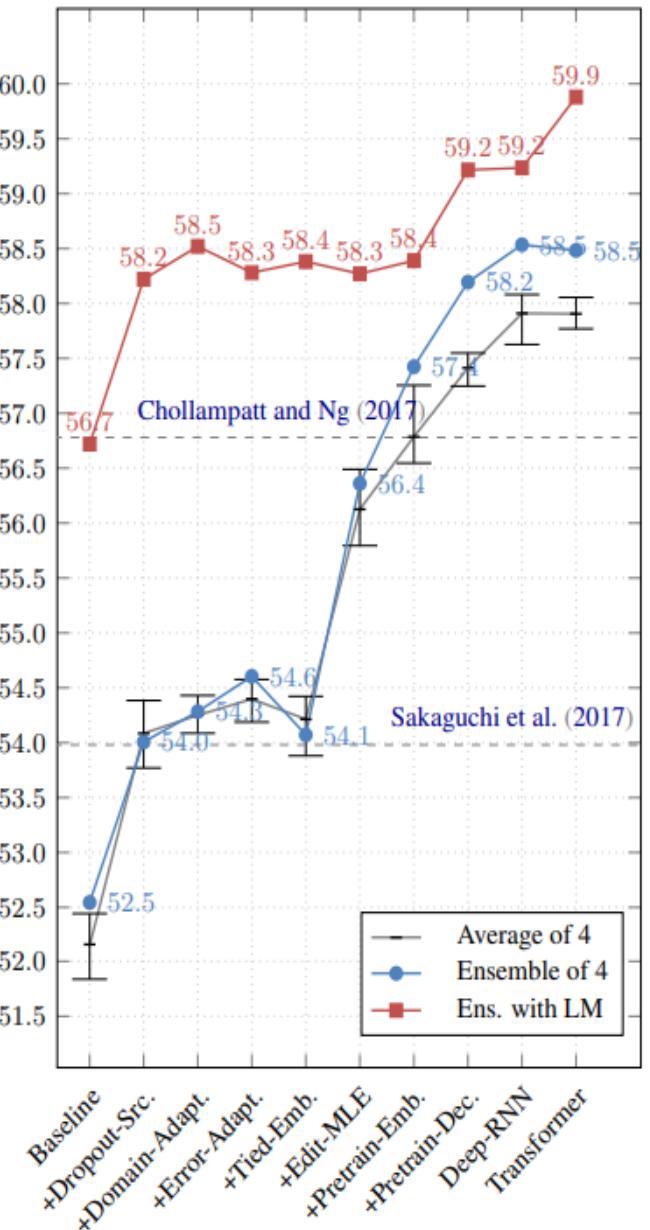
Σε προηγούμενες εργασίες, οι νευρωνικές μέθοδοι στη διόρθωση γραμματικών λαθών (GEC) δεν έφτασαν σε αποτελέσματα state-of-the art σε σύγκριση με τις βασικές μεθόδους στατιστικής μηχανικής μετάφρασης (SMT) που βασίζονται σε φράσεις. Έτσι λοιπόν, αυτή η εργασία επιδεικνύει παραλληλισμούς μεταξύ της νευρωνικής GEC και της low-resource νευρωνικής MT, και προσαρμόζει με επιτυχία διάφορες μεθόδους από τη low-resource MT στη νευρωνική GEC. Καθορίζει περαιτέρω κατευθυντήριες γραμμές για αξιόπιστα αποτελέσματα στη νευρωνική GEC και προτείνει ένα σύνολο ανεξάρτητων από μοντέλα μεθόδων για τη νευρωνική GEC που μπορούν εύκολα να εφαρμοστούν στις περισσότερες ρυθμίσεις της. Οι προτεινόμενες μέθοδοι περιλαμβάνουν την προσθήκη θορύβου από την πλευρά της πηγής, τεχνικές προσαρμογής στον χώρο, έναν ειδικό στόχο εκπαίδευσης για τη GEC, μάθηση μεταφοράς (transfer learning) με μονόγλωσσα δεδομένα και συνένωση ανεξάρτητα εκπαίδευμένων μοντέλων GEC με γλωσσικά μοντέλα. Οι συνδυασμένες επιδράσεις αυτών των μεθόδων οδηγούν σε καλύτερα state-of-the art νευρωνικά μοντέλα GEC που ξεπερνούν τα προηγουμένως καλύτερα νευρωνικά συστήματα GEC κατά περισσότερο από 10%, χρησιμοποιώντας τη μετρική  $M^2$ , στο CoNLL-2014 benchmark και κατά 5,9% στο σύνολο δοκιμών JFLEG. Τα μη νευρωνικά state-of-the art συστήματα υπερτερούν κατά περισσότερο από 2% στο benchmark CoNLL-2014 και κατά 4% στο JFLEG [12].

Model	Dev	Prec.	Rec.	Test
Baseline	19.3	70.8	9.5	30.9
+Dropout-Src.	27.5	72.4	15.5	41.7
+Domain-Adapt.	30.0	69.2	17.3	43.3
+Error-Adapt.	34.5	70.8	20.8	47.8
+Tied-Emb.	33.0	73.0	20.2	48.0
+Edit-MLE	37.6	65.3	27.1	51.0

Εικόνα 2.24: Αποτελέσματα ( $M^2$ ) στο benchmark CoNLL για ειδικές προσαρμογές για τη GEC.



(a) CoNLL-2014 test set ( $M^2$ )

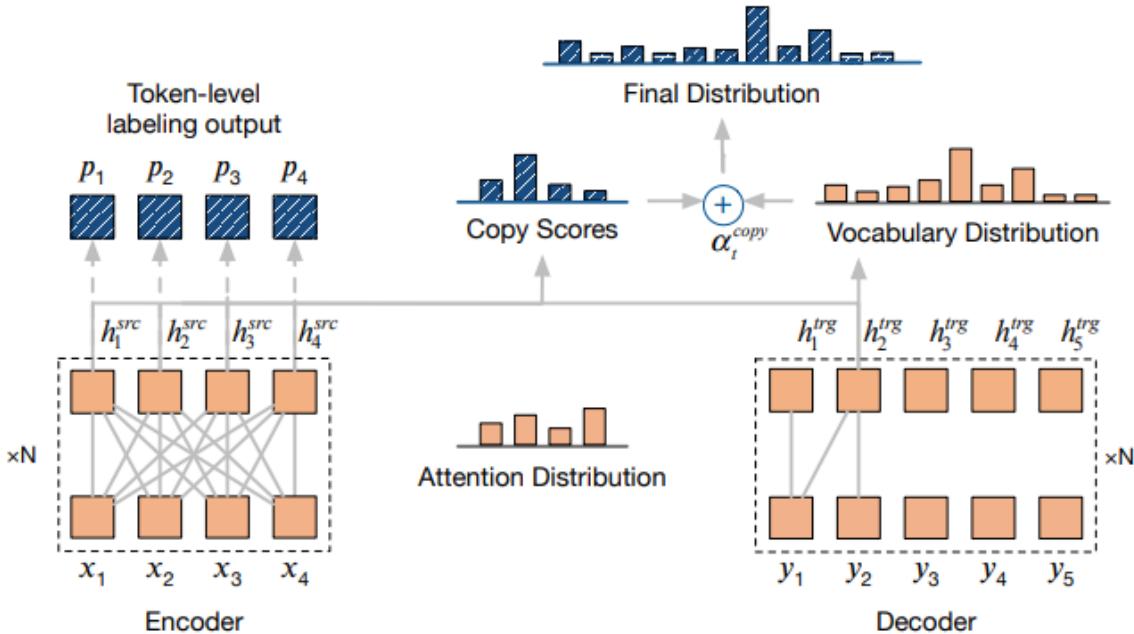


(b) JFLEG test set (GLEU)

Εικόνα 2.25: Σύγκριση στο σύνολο δοκιμών CoNLL-2014 και στο σύνολο δοκιμών JFLEG για όλες τις εξεταζόμενες μεθόδους στην εργασία

## Zhao (2019)

Τα νευρωνικά συστήματα αυτόματης μετάφρασης έχουν γίνει οι πιο σύγχρονες προσεγγίσεις για την εργασία διόρθωσης γραμματικών σφαλμάτων (GEC). Στην παρούσα εργασία, προτείνεται μια αρχιτεκτονική copy-augmented για την εργασία GEC, αντιγράφοντας τις αμετάβλητες λέξεις από την πρόταση-πηγή στην πρόταση-στόχο, δεδομένου ότι η GEC πάσχει από τη μη ύπαρξη αρκετών επισημειωμένων δεδομένων εκπαίδευσης για την επίτευξη υψηλής ακρίβειας. Εκπαιδεύεται εκ των προτέρων η αρχιτεκτονική copy-augmented με έναν αυτόματο κωδικοποιητή αποθορυβοποίησης χρησιμοποιώντας το μη επισημασμένο One Billion Benchmark και γίνονται συγκρίσεις μεταξύ του πλήρως προ-εκπαίδευμένου μοντέλου και ενός μερικώς προ-εκπαίδευμένου μοντέλου. Είναι η πρώτη φορά που η αντιγραφή λέξεων από το αρχικό πλαίσιο και η πλήρης προεκπαίδευση ενός μοντέλου sequence-to-sequence πειραματίζονται στην εργασία GEC. Επιπλέον, προστίθεται token-level και sentence-level multi-task μάθηση για την εργασία GEC. Τα αποτελέσματα αξιολόγησης στο σύνολο δοκιμών CoNLL-2014 δείχνουν ότι η προσέγγιση αυτή ξεπερνά με μεγάλη διαφορά όλα τα πρόσφατα δημοσιευμένα αποτελέσματα της τελευταίας τεχνολογίας. Ο κώδικας και τα προ-εκπαίδευμένα μοντέλα έχουν δημοσιευθεί στο [Github](#) [13].



Εικόνα 2.26: Copy-Augmented αρχιτεκτονική

Model	Year	CoNLL-14			JFELEG GLEU	Dict
		Pre.	Rec.	$F_{0.5}$		
SMT (with LM)	2014	41.72	22.00	35.38	-	word
SMT Rule-Based Hybird (with LM)	2014	39.71	30.10	37.33	-	word
SMT Classification Hybird (with LM)	2016	60.17	25.64	47.40	-	word
Neural Hybird MT (with LM)	2017	-	-	45.15	53.41	char/word
CNN + EO (4 ens. with LM)	2018	65.49	33.14	54.79	57.47	bpe
Transformer + MIMs (4 ens. with LM)	2018	63.00	38.90	56.10	59.90	bpe
NMT SMT Hybrid (4 ens. with LM)	2018	66.77	34.49	56.25	61.50	bpe
<b>Our Model</b>						
Copy-augmented Model (4 ens.)	-	68.48	33.10	<b>56.42</b>	59.48*	word
+ DA, Multi-tasks (4 ens.)	-	71.57	38.65	<b>61.15</b>	61.00*	word
<b>Model Trained with Large Non-public Training Data</b>						
CNN + FB Learning (4 ens. with LM)	2018	74.12	36.30	61.34	61.41	bpe

Εικόνα 2.27: Σύγκριση των συστημάτων GEC στο σύνολο δοκιμών CoNLL-2014 και JFLEG. Αναφέρεται η βαθμολογία  $M^2$  για το σύνολο δοκιμαστικών δεδομένων CoNLL-2014 και η βαθμολογία GLEU για το σύνολο δοκιμαστικών δεδομένων JFLEG. Το DA αναφέρεται στον "Αυτόματο κωδικοποιητή αποθρυβοποίησης" (Denoising Auto-encoder). Η ένδειξη (with LM) αναφέρεται στη χρήση ενός επιπλέον γλωσσικού μοντέλου. Η ένδειξη (4 ens.) αναφέρεται στην ensemble αποκωδικοποίηση 4 ανεξάρτητα εκπαιδευμένων μοντέλων. Επανακατατάσσονται τα αποτελέσματα των 12 κορυφαίων υποθέσεων για το σύνολο δοκιμών JFLEG με ένα επιπλέον γλωσσικό μοντέλο και σημειώνονται με \*

Model	Pre.	Rec.	$F_{0.5}$	Imp.
Transformer	55.96	30.73	48.07	-
+ Copying	65.23	33.18	<b>54.67</b>	+6.60
<b>Ignoring UNK words as edits</b>				
Transformer	65.26	30.63	53.23	-
+ Copying	65.54	33.18	54.85	+1.62
<b>+ Pre-training</b>				
Copy-Augmented Transformer	65.23	33.18	54.67	-
+ Pre-training Decoder (partially pre-trained)	68.02	34.98	<b>57.21</b>	+2.54
+ Denosing Auto-encoder (fully pre-trained)	68.97	36.98	<b>58.80</b>	+4.13
<b>+ Multi-tasks</b>				
Copy-Augmented Transformer	67.74	40.62	<b>59.76</b>	-

Εικόνα 2.28: Μελέτη κατάλυσης (ablation study) με μονωμένων μοντέλων στο σύνολο δεδομένων δοκιμής CoNLL-2014

## Kiyono (2019)

Η ενσωμάτωση ψευδοδεδομένων (pseudo data) στην εκπαίδευση των μοντέλων διόρθωσης γραμματικών λαθών υπήρξε ένας από τους κύριους παράγοντες για τη βελτίωση της απόδοσης των εν λόγω μοντέλων. Ωστόσο, δεν υπάρχει συναίνεση όσον αφορά τις πειραματικές διαμορφώσεις, δηλαδή την επιλογή του τρόπου με τον οποίο θα πρέπει να δημιουργούνται ή να χρησιμοποιούνται τα ψευδοδεδομένα. Στην παρούσα μελέτη, οι επιλογές αυτές διερευνώνται μέσω εκτεταμένων πειραμάτων και επιτυγχάνονται επιδόσεις τελευταίας τεχνολογίας στο σύνολο δοκιμών CoNLL-2014 ( $F_{0.5} = 65.0$ ) και στο επίσημο σύνολο δοκιμών της διαμοιρασμένης εργασίας BEA-2019 ( $F_{0.5} = 70.2$ ) χωρίς να γίνουν τροποποιήσεις στην αρχιτεκτονική του μοντέλου. Τα ψευδοδεδομένα δημιουργούνται με 3 μεθόδους που ονομάζονται BACKTRANS (NOISY), BACKTRANS (SAMPLE) και DIRECTNOISE [14].

Model	Ensemble	CoNLL-2014 ( $M^2$ scorer)			CoNLL-2014 (ERRANT)			JFLEG	BEA-test (ERRANT)		
		Prec.	Rec.	$F_{0.5}$	Prec.	Rec.	$F_{0.5}$		GLEU	Prec.	Rec.
Chollampatt and Ng (2018)		60.9	23.7	46.4	-	-	-	51.3	-	-	-
Junczys-Dowmunt et al. (2018)		-	-	53.0	-	-	-	57.9	-	-	-
Grundkiewicz and Junczys-Dowmunt (2018)		66.8	34.5	56.3	-	-	-	61.5	-	-	-
Lichtarge et al. (2019)		65.5	37.1	56.8	-	-	-	61.6	-	-	-
Chollampatt and Ng (2018)	✓	65.5	33.1	54.8	-	-	-	57.5	-	-	-
Junczys-Dowmunt et al. (2018)	✓	61.9	40.2	55.8	-	-	-	59.9	-	-	-
Lichtarge et al. (2019)	✓	66.7	43.9	60.4	-	-	-	<b>63.3</b>	-	-	-
Zhao et al. (2019)	✓	71.6	38.7	61.2	-	-	-	61.0	-	-	-
Grundkiewicz et al. (2019)	✓	-	-	64.2	-	-	-	61.2	72.3	60.1	69.5
PRETLARGE		67.9	44.1	61.3	61.2	42.0	56.0	59.7	65.5	59.4	64.2
PRETLARGE+SSE+R2L	✓	72.4	<b>46.1</b>	<b>65.0</b>	67.3	<b>44.0</b>	<b>60.9</b>	61.4	72.1	<b>61.8</b>	69.8
PRETLARGE+SSE+R2L+SED	✓	<b>73.3</b>	44.2	64.7	<b>68.1</b>	42.1	60.6	61.2	<b>74.7</b>	56.7	<b>70.2</b>

Εικόνα 2.29: Σύγκριση του καλύτερου μοντέλου της εργασίας και των κορυφαίων μοντέλων της εποχής: μια **bold** τιμή υποδηλώνει το καλύτερο αποτέλεσμα εντός της στήλης.

Dataset	#sent (pairs)	#refs.	Split	Scorer
BEA-train	561,410	1	train	-
BEA-valid	2,377	1	valid	ERRANT
CoNLL-2014	1,312	2	test	ERRANT & $M^2$ scorer
JFLEG	1,951	4	test	GLEU
BEA-test	4,477	5	test	ERRANT
SimpleWiki*	1,369,460	-	-	-
Wikipedia*	145,883,941	-	-	-
Gigaword*	131,864,979	-	-	-

Εικόνα 2.30: Περίληψη των συνόλων δεδομένων που χρησιμοποιήθηκαν στα πειράματα της εργασίας. Τα σύνολα δεδομένων που επισημαίνονται με "\*" είναι ένα αρχικό σώμα δεδομένων  $T$ .

Method	Prec.	Rec.	$F_{0.5}$
Baseline	46.6	23.1	38.8
BACKTRANS (SAMPLE)	44.6	27.4	39.6
BACKTRANS (NOISY)	42.5	<b>31.3</b>	39.7
DIRECTNOISE	<b>48.9</b>	25.7	<b>41.4</b>

Εικόνα 2.31: Απόδοση των μοντέλων στο BEA-valid. Μια τιμή με **bold** υποδηλώνει το καλύτερο αποτέλεσμα στη στήλη. Το αρχικό σώμα  $T$  είναι το SimpleWiki

## Kaneko (2020)

Η παρούσα εργασία διερευνά τον τρόπο αποτελεσματικής ενσωμάτωσης ενός προ-εκπαιδευμένου μοντέλου masked language (MLM), όπως το BERT, σε ένα μοντέλο κωδικοποιητή-αποκωδικοποιητή (EncDec) για τη διόρθωση γραμματικών λαθών (GEC). Η απάντηση σε αυτό το ερώτημα δεν είναι τόσο απλή όσο θα περίμενε κανείς, διότι οι προηγούμενες κοινές μέθοδοι για την ενσωμάτωση ενός MLM σε ένα μοντέλο EncDec έχουν πιθανά μειονεκτήματα όταν εφαρμόζονται στην GEC. Για παράδειγμα, η κατανομή των εισόδων σε ένα μοντέλο GEC μπορεί να είναι σημαντικά διαφορετική (λανθασμένη, αδέξια κ.λπ.) από εκείνη των σωμάτων που χρησιμοποιούνται για την προ-εκπαίδευση των MLM. Ωστόσο, το ζήτημα αυτό δεν αντιμετωπίζεται στις προηγούμενες μεθόδους. Τα πειράματα της εργασίας αυτής δείχνουν ότι η προτεινόμενη μέθοδος, όπου πρώτα τελειοποιείται (fine-tuning) ένα MLM με ένα δεδομένο σώμα GEC και στη συνέχεια χρησιμοποιείται η έξοδός του ως πρόσθετα χαρακτηριστικά στο μοντέλο GEC, μεγιστοποιεί το όφελος του MLM. Το μοντέλο με τις καλύτερες επιδόσεις επιτυγχάνει state-of-the-art επιδόσεις στα benchmark BEA-2019 και CoNLL-2014. Ο κώδικας της εργασίας είναι δημόσια διαθέσιμος στο [Github](#) [15].

	BEA-test (ERRANT)			CoNLL-14 (M <sup>2</sup> )			FCE-test (M <sup>2</sup> )			JFLEG
	P	R	F <sub>0.5</sub>	P	R	F <sub>0.5</sub>	P	R	F <sub>0.5</sub>	GLEU
w/o BERT	51.5	43.2	49.6	59.2	31.2	50.2	61.7	46.4	57.9	52.7
BERT-init	55.1	43.7	52.4	61.3	31.5	51.4	62.4	46.9	58.5	53.0
BERT-fuse	57.5	<b>44.9</b>	54.4	62.3	31.3	52.0	64.0	47.6	59.8	54.1
BERT-fuse mask	57.1	44.7	54.1	62.9	32.2	52.8	64.3	48.1	60.2	54.2
BERT-fuse GED	<b>58.1</b>	44.8	<b>54.8</b>	<b>63.6</b>	<b>33.0</b>	<b>53.6</b>	<b>65.0</b>	<b>49.6</b>	<b>61.2</b>	<b>54.4</b>
w/o BERT	66.1	59.9	64.8	68.5	44.8	61.9	56.5	48.1	54.9	61.0
BERT-fuse	66.6	60.0	65.2	68.3	<b>45.7</b>	62.1	59.7	<b>48.5</b>	<b>57.0</b>	61.2
BERT-fuse mask	67.0	60.0	65.4	68.8	45.3	62.3	59.7	47.1	56.6	61.2
BERT-fuse GED	<b>67.1</b>	<b>60.1</b>	<b>65.6</b>	<b>69.2</b>	45.6	<b>62.6</b>	<b>59.8</b>	46.9	56.7	61.3
Lichtarge et al. (2019)	-	-	-	65.5	37.1	56.8	-	-	-	<b>61.6</b>
Awasthi et al. (2019)	-	-	-	66.1	43.0	59.7	-	-	-	60.3
Kiyono et al. (2019)	65.5	59.4	64.2	67.9	44.1	61.3	-	-	-	59.7
BERT-fuse GED + R2L	72.3	<b>61.4</b>	69.8	<b>72.6</b>	<b>46.4</b>	<b>65.2</b>	62.8	48.8	59.4	62.0
Lichtarge et al. (2019)	-	-	-	66.7	43.9	60.4	-	-	-	<b>63.3</b>
Grundkiewicz et al. (2019)	72.3	60.1	69.5	-	-	64.2	-	-	-	61.2
Kiyono et al. (2019)*	<b>74.7</b>	56.7	<b>70.2</b>	72.4	46.1	65.0	-	-	-	61.4

Εικόνα 2.32: Αποτελέσματα των μοντέλων GEC. Το πρώτο γκρουπ παρουσιάζει τα αποτελέσματα των μεμονωμένων μοντέλων χωρίς τη χρήση ψευδοδεδομένων ή/και ensemble. Το δεύτερο γκρουπ δείχνει τα αποτελέσματα των μεμονωμένων μοντέλων με χρήση ψευδοδεδομένων. Το τρίτο γκρουπ παρουσιάζει τα μοντέλα ensemble με χρήση ψευδοδεδομένων. Τα **bold** γράμματα υποδηλώνουν την υψηλότερη βαθμολογία σε κάθε στήλη. Η ένδειξη \* αναφέρει τα state-of-the-art αποτελέσματα για το τεστ BEA και το CoNLL-2014 για δύο ξεχωριστά μοντέλα: μοντέλα με και χωρίς SED

---

## GEC model

---

Model Architecture	Transformer (big)
Number of epochs	30
Max tokens	4096
Optimizer	Adam $(\beta_1 = 0.9, \beta_2 = 0.98, \epsilon = 1 \times 10^{-8})$
Learning rate	$3 \times 10^{-5}$
Min learning rate	$1 \times 10^{-6}$
Loss function	label smoothed cross-entropy $(\epsilon_{ls} = 0.1)$ <a href="#">(Szegedy et al., 2016)</a>
Dropout	0.3
Gradient Clipping	0.1
Beam search	5

---

## GED model

---

Model Architecture	BERT-Base (cased)
Number of epochs	3
Batch size	32
Max sentence length	128
Optimizer	Adam $(\beta_1 = 0.9, \beta_2 = 0.999, \epsilon = 1 \times 10^{-8})$
Learning rate	$4e - 5$
Dropout	0.1

---

*Εικόνα 2.33: Τιμές υπερπαραμέτρων του μοντέλου GEC και του Fine-tuned BERT*

## Omelianchuk (2020)

Στην παρούσα εργασία, παρουσιάζεται ένας απλός και αποτελεσματικός επισημειωτής ακολουθιών (sequence tagger) GEC που χρησιμοποιεί έναν κωδικοποιητή Transformer. Το σύστημα προ-εκπαιδεύεται σε συνθετικά δεδομένα και στη συνέχεια τελειοποιείται σε δύο στάδια. Πρώτα σε σώματα με λάθη και στη συνέχεια σε ένα συνδυασμό από παράλληλα σώματα με λάθη και χωρίς λάθη. Σχεδιάζονται token-level προσαρμοσμένοι μετασχηματισμοί για να αντιστοιχίσουμε τα tokens εισόδου σε διορθώσεις-στόχους. Ο καλύτερος GEC tagger ενός μοντέλου/συνόλου επιτυγχάνει  $F_{0.5}$  65.3/66.5 στο CoNLL-2014 (test) και  $F_{0.5}$  72.4/73.6 στο BEA-2019 (test). Η ταχύτητα εξαγωγής συμπερασμάτων είναι έως και 10 φορές ταχύτερη από ένα σύστημα GEC με βάση τον Transformer seq2seq. Ο κώδικας και τα εκπαιδευμένα μοντέλα είναι δημόσια διαθέσιμα στο [Github](#) [16].

GEC system	Ens.	CoNLL-2014 (test)			BEA-2019 (test)		
		P	R	$F_{0.5}$	P	R	$F_{0.5}$
Zhao et al. (2019)		67.7	40.6	59.8	-	-	-
Awasthi et al. (2019)		66.1	43.0	59.7	-	-	-
Kiyono et al. (2019)		67.9	<b>44.1</b>	61.3	65.5	<b>59.4</b>	64.2
Zhao et al. (2019)	✓	74.1	36.3	61.3	-	-	-
Awasthi et al. (2019)	✓	68.3	43.2	61.2	-	-	-
Kiyono et al. (2019)	✓	72.4	<b>46.1</b>	65.0	74.7	56.7	70.2
Kantor et al. (2019)	✓	-	-	-	78.3	58.0	73.2
GECToR (BERT)		72.1	42.0	63.0	71.5	55.7	67.6
GECToR (RoBERTa)		73.9	41.5	64.0	77.2	55.1	71.5
GECToR (XLNet)		<b>77.5</b>	40.1	<b>65.3</b>	<b>79.2</b>	53.9	<b>72.4</b>
GECToR (RoBERTa + XLNet)	✓	76.6	42.3	66.0	<b>79.4</b>	57.2	<b>73.7</b>
GECToR (BERT + RoBERTa + XLNet)	✓	<b>78.2</b>	41.5	<b>66.5</b>	78.9	<b>58.2</b>	73.6

Εικόνα 2.34: Σύγκριση μεμονωμένων μοντέλων και συνόλων. Το  $M^2$  για το CoNLL-2014 (test) και ERRANT για το BEA-2019 (test) αναφέρονται. Στα σύνολα απλά υπολογίζουμε τον μέσο όρο των πιθανοτήτων εξόδου από τα μεμονωμένα μοντέλα.

## Rothe (2021)

Η παρούσα εργασία παρουσιάζει μια απλή φόρμουλα για την εκπαίδευση σύγχρονων πολύγλωσσων μοντέλων διόρθωσης γραμματικών σφαλμάτων (GEC). Αυτό επιτυγχάνεται προτείνοντας αρχικά μια μέθοδο που δεν επηρεάζει τη γλώσσα για τη δημιουργία μεγάλου αριθμού συνθετικών παραδειγμάτων. Το δεύτερο συστατικό είναι η χρήση πολύγλωσσων γλωσσικών μοντέλων μεγάλης κλίμακας (έως και 11B παράμετροι). Με τη λεπτομερή ρύθμιση σε σύνολα εποπτευόμενων γλωσσών, η εργασία ξεπερνά τα προηγούμενα state-of-the-art αποτελέσματα σε δείκτες αναφοράς GEC σε τέσσερις γλώσσες: Αγγλικά, Τσεχικά, Γερμανικά και Ρωσικά. Έχοντας καθιερωθεί ένα νέο σύνολο βασικών γραμμών για το GEC, τα αποτελέσματά μας γίνονται εύκολα αναπαραγόμενα και προσβάσιμα με τη δημοσίευση ενός συνόλου δεδομένων CLANG-8. Παράγεται από τη χρήση του καλύτερου μοντέλου της εργασίας, το οποίο ονομάζεται gT5, για τον καθαρισμό των στόχων ενός ευρέως χρησιμοποιούμενου αλλά θορυβώδους συνόλου δεδομένων LANG-8. Το CLANG-8 απλοποιεί σημαντικά τις τυπικές σωληνώσεις εκπαίδευσης GEC, που αποτελούνται από πολλαπλά στάδια τελειοποίησης. Αποδεικνύεται ότι η εκτέλεση ενός μόνο βήματος τελειοποίησης (fine-tuning) στο CLANG-8 με τα έτοιμα γλωσσικά μοντέλα, αποδίδει περαιτέρω βελτίωση της ακρίβειας σε σχέση με ένα ήδη κορυφαίο μοντέλο gT5 για την αγγλική γλώσσα [17].

Models	<i>CoNLL-14</i>	<i>BEA test</i>	Czech	German	Russian
Omelianchuk et al.*	66.5	<b>73.6</b>	-	-	-
Lichtarge et al.*	<b>66.8</b>	73.0	-	-	-
Náplava and Straka	63.40	69.00	80.17	73.71	50.20
Katsumata and Komachi*	63.00	66.10	73.52	68.86	44.36
gT5 base	54.10	60.2	71.88	69.21	26.24
gT5 xxl	65.65	69.83	<b>83.15</b>	<b>75.96</b>	<b>51.62</b>

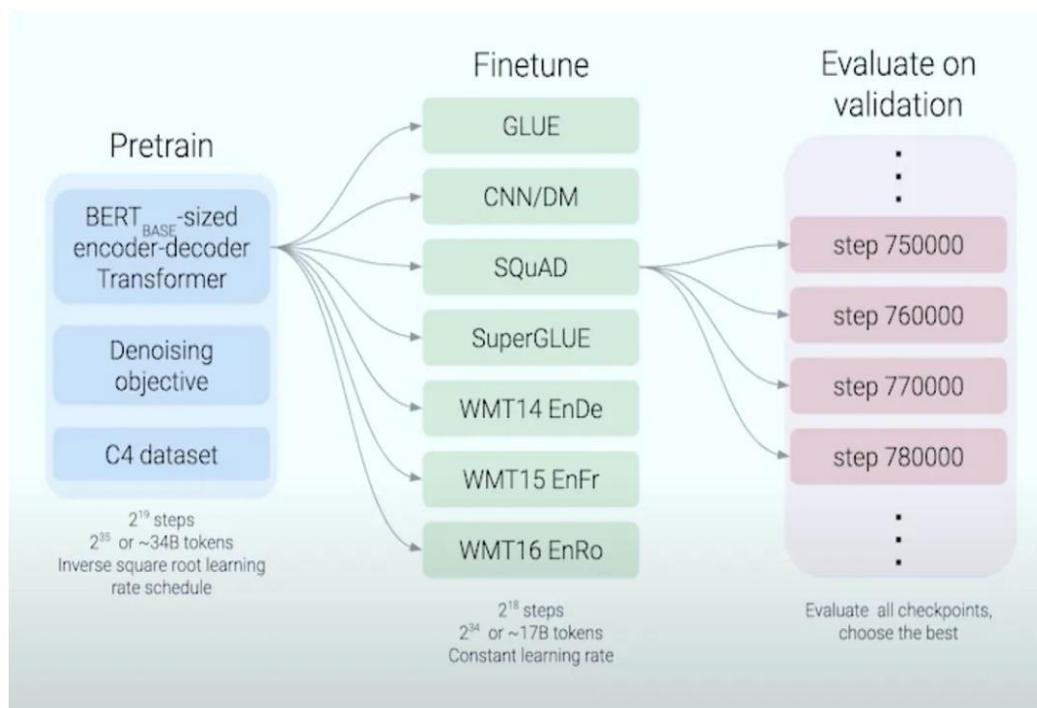
Εικόνα 2.35:  $F_{0.5}$  Scores. Τα μοντέλα που συμβολίζονται με \* είναι μοντέλα συνόλου. Χρησιμοποιήσαμε την  $M^2$  για το CoNLL-14, Ρωσικά, Τσεχικά και Γερμανικά, και το ERRANT scorer για το τεστ BEA

Model	#params	Training Data	<i>CoNLL-14</i>	<i>BEA test</i>
SOTA			66.8	73.6
gT5 xxl			65.65	69.83
FELIX	220M	LANG-8	41.63	30.54
FELIX	220M	LANG-8 + BEA	48.75	48.80
FELIX	220M	cLANG-8	<b>58.21</b>	<b>59.05</b>
T5 base	220M	LANG-8	52.77	59.14
T5 base	220M	LANG-8 + BEA	60.61	67.12
T5 base	220M	cLANG-8	<b>65.13</b>	<b>69.38</b>
T5 base	220M	cLANG-8-S	58.70	59.95
T5 small	60M	cLANG-8	60.70	65.01
T5 base	220M	cLANG-8	65.13	69.38
T5 large	770M	cLANG-8	66.10	72.06
T5 xl	3B	cLANG-8	67.75	73.92
T5 xxl	11B	cLANG-8	<b>68.87</b>	<b>75.88</b>

Εικόνα 2.36: Βαθμολογία  $F_{0.5}$  στο τεστ CoNLL-14 και στο τεστ BEA. Τα μπλοκ δύο και τρία συγκρίνουν διαφορετικά δεδομένα εκπαίδευσης. Το τελευταίο μπλοκ συγκρίνει διαφορετικά μεγέθη μοντέλων για το μοντέλο T5.

### 3. Μεθοδολογία Εκπαίδευσης Μοντέλου

Σε αυτή την ενότητα θα αναδείξουμε την τεχνική του πως να εκπαιδεύσουμε ένα μοντέλο Transformer τελευταίας τεχνολογίας για να εκτελεί διόρθωση γραμματικών λαθών. Θα χρησιμοποιήσουμε ένα μοντέλο που ονομάζεται T5 (Text-to-Text Transfer Transformer), το οποίο επί του παρόντος ξεπερνάει την ανθρώπινη βάση στο μέτρο σύγκρισης General Language Understanding Evaluation (GLUE), καθιστώντας το ένα από τα πιο ισχυρά μοντέλα NLP που υπάρχουν [18]. Το T5 δημιουργήθηκε από την Google AI και κυκλοφόρησε στον κόσμο ώστε να είναι προσβάσιμο και ελεύθερο προς χρήση από τον καθένα.



Εικόνα 3.1: Τα βήματα προεκπαίδευσης, λεπτομερούς ρύθμισης και αξιολόγησης

#### 3.1 Συμβατά Dataset

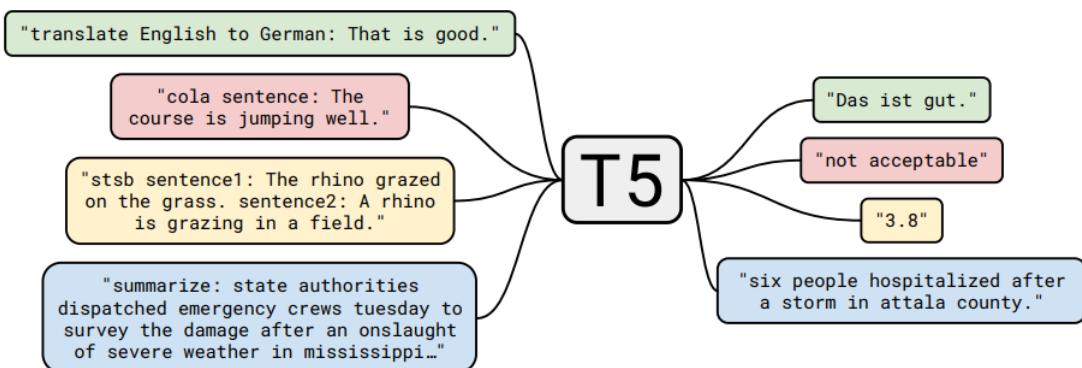
Το συγκεκριμένο μοντέλο μπορεί να δουλέψει με πολλών ειδών dataset για τις εργασίες που επιτελεί. Εκτός από το καθαρό σύνολο δεδομένων C4, έχουν δοκιμαστεί επίσης τα ίδια δεδομένα χωρίς κανένα φιλτράρισμα. Το αποτέλεσμα δείχνει ότι το φιλτράρισμα βιοθά το μοντέλο να αποδώσει καλύτερα. Εφαρμόζονται επίσης και άλλα σύνολα δεδομένων με μικρότερη τάξη μεγέθους. Τα αποτελέσματα δείχνουν ότι η προεκπαίδευση σε σύνολα δεδομένων βιοθά στην απόδοση των επόμενων εργασιών. Για παράδειγμα, το προ-εκπαίδευμένο μοντέλο T5 σε σύνολο δεδομένων που μοιάζει με το RealNews είχε πολύ καλύτερες επιδόσεις [19].

Dataset	Size	GLUE	CNNMD
★ C4	745GB	83.28	<b>19.24</b>
C4, unfiltered	6.1TB	81.46	19.14
RealNews-like	35GB	<b>83.83</b>	<b>19.23</b>
WebText-like	17GB	<b>84.03</b>	<b>19.31</b>
Wikipedia	16GB	81.85	<b>19.31</b>
Wikipedia + TBC	20GB	83.65	<b>19.28</b>

Much worse on CoLA  
Orders of magnitude smaller

Εικόνα 3.2: Διαφορετικά σύνολα δεδομένων για το μοντέλο και οι επιδόσεις τους

Η βασική ιδέα της εργασίας είναι να αντιμετωπιστεί κάθε πρόβλημα επεξεργασίας κειμένου ως ένα πρόβλημα text-to-text, δηλαδή λαμβάνοντας κείμενο ως είσοδο και παράγοντας νέο κείμενο ως έξοδο [20]. Αυτή η προσέγγιση είναι εμπνευσμένη από προηγούμενα ενοποιημένα πλαίσια για NLP εργασίες, συμπεριλαμβανομένης της μετατροπής όλων των προβλημάτων κειμένου σε εργασίες απάντησης ερωτήσεων, γλωσσικής μοντελοποίησης ή εξαγωγής διαστήματος. Είναι σημαντικό ότι το πλαίσιο text-to-text μας επιτρέπει να εφαρμόζουμε άμεσα το ίδιο μοντέλο, το ίδιο αντικείμενο, την ίδια διαδικασία εκπαίδευσης και την ίδια διαδικασία αποκωδικοποίησης σε κάθε έργο που εξετάζουμε. Με αυτή την ενιαία προσέγγιση, μπορούμε να συγκρίνουμε την αποτελεσματικότητα διαφορετικών αντικειμένων transfer learning, μη επισημειωμένων συνόλων δεδομένων και άλλων παραγόντων, ενώ παράλληλα διερευνούμε τα όρια του transfer learning για NLP με την επέκταση των μοντέλων και των συνόλων δεδομένων πέρα από αυτά που έχουν εξεταστεί προηγουμένως.



Εικόνα 3.3: Ένα διάγραμμα του πλαισίου μας για το μετασχηματισμό text-to-text. Κάθε εργασία που εξετάζουμε, συμπεριλαμβανομένων της μετάφρασης, της απάντησης ερωτήσεων και της ταξινόμησης είναι η τροφοδοσία του μοντέλου μας με κείμενο ως είσοδο και η εκπαίδευσή του ώστε να παράγει κάποιο κείμενο στόχο.

## 3.2 Ανάλυση Λειτουργίας Μοντέλου

Προκειμένου να εκπαιδεύσουμε ένα ενιαίο μοντέλο στο ευρύ σύνολο εργασιών που περιγράφηκε παραπάνω, μετατρέπουμε όλες τις εργασίες που εξετάζουμε σε μια μορφή text-to-text. Αυτό το πλαίσιο παρέχει έναν συνεπή στόχο εκπαίδευσης τόσο για την προ-εκπαίδευση όσο και για τη λεπτομερή ρύθμιση. Συγκεκριμένα, το μοντέλο εκπαιδεύεται με ένα σκοπό μέγιστης πιθανότητας ανεξάρτητα από την εργασία. Για να προσδιορίσουμε ποια εργασία πρέπει να εκτελέσει το μοντέλο, προσθέτουμε ένα ειδικό για την εργασία πρόθεμα στην αρχική ακολουθία εισόδου πριν την τροφοδοτήσουμε στο μοντέλο.

```
example = "grammar: This sentences, has bads grammar and spelling!"
```

Εικόνα 3.4: Παράδειγμα εισόδου στο μοντέλο, που συνοδεύεται από το ειδικό πρόθεμα για διόρθωση γραμματικών λαθών (grammar).

Η εργασία μας διενεργείται στο περιβάλλον Google Colab [21]. Το περιβάλλον αυτό, χρησιμοποιεί gpus (T4 GPU) για την επιτάχυνση του υλικού. Αυτό σημαίνει πως όλες οι διεργασίες γίνονται ταχύτατα, κάτι που μας βοηθά λόγω του μεγάλου όγκου δεδομένων που χρειάζεται να παραχθούν για την εργασία. Για την υλοποίηση της εργασίας μας θα χρησιμοποιήσουμε το πακέτο Python που ονομάζεται Happy Transformer [22]. Το Happy Transformer είναι χτισμένο πάνω στη βιβλιοθήκη Transformers του Hugging Face [23], διευκολύνει τη λεπτομερή ρύθμιση και την εξαγωγή συμπερασμάτων με τα μοντέλα της NLP Transformer και καθιστά εύκολη την υλοποίηση και την εκπαίδευση μοντέλων μετασχηματιστών με λίγες γραμμές κώδικα. Έτσι, δεν απαιτείται καμία σύνθετη κατανόηση της NLP ή της Python, παρόλο που θα εκπαιδεύσουμε ένα από τα πιο ικανά μοντέλα τεχνητής νοημοσύνης στον κόσμο [24].

Public Methods	Basic Usage	Training
Text Generation	✓	✓
Text Classification	✓	✓
Question Answering	✓	✓
Word Prediction	✓	✓
Text-to-Text	✓	✓
Token Classification	✓	
Next Sentence Prediction	✓	

Εικόνα 3.5: Χαρακτηριστικά του Happy Transformer

Αφού εγκαταστήσουμε το Happy Transformer, εισάγουμε την κλάση HappyTextToText την οποία θα χρησιμοποιήσουμε για να φορτώσουμε το μοντέλο και την κλάση δεδομένων με τους τύπους πεδίων (TTTrainArgs) που θέλουμε να χρησιμοποιήσουμε.

```
from happytransformer import HappyTextToText  
from happytransformer import TTTrainArgs
```

Εικόνα 3.6: Εισαγωγή HappyTextToText και κλάσης δεδομένων στο πρόγραμμα

Οι παράμετροι που περιέχονται μέσα στην κλάση δεδομένων είναι οι εξής:

- ✓ **learning\_rate**: Πόσο προσαρμόζονται τα βάρη του μοντέλου ανά βήμα. Πολύ χαμηλό και το μοντέλο θα χρειαστεί πολύ χρόνο για να μάθει ή θα κολλήσει σε μια μη βέλτιστη λύση. Πολύ υψηλό μπορεί να προκαλέσει πιθανές αποκλίνουσες συμπεριφορές.
- ✓ **num\_train\_epochs**: Ο αριθμός των επαναλήψεων (iterations) των δεδομένων εκπαίδευσης.
- ✓ **batch\_size**: Ο αριθμός των παραδειγμάτων εκπαίδευσης (training examples) που χρησιμοποιούνται ανά επανάληψη.
- ✓ **preprocessing\_processes**: Αριθμός διεργασιών που χρησιμοποιούνται για την προεπεξεργασία. Συνιστώνται 1 έως 4.
- ✓ **save\_path**: Διαδρομή προς ένα φάκελο για την αποθήκευση των δεδομένων.
- ✓ **load\_path**: Διαδρομή προς ένα φάκελο για τη φόρτωση των δεδομένων
- ✓ **max\_input\_length**: Ο μέγιστος αριθμός tokens για την είσοδο. Τα υπόλοιπα περικόπτονται. Από προεπιλογή χρησιμοποιείται ο μέγιστος αριθμός tokens που μπορεί να διαχειριστεί το μοντέλο.
- ✓ **max\_output\_length**: Το ίδιο με την παράμετρο **max\_input\_length**, εκτός από την έξοδο.
- ✓ **fp16**: Αν είναι `true`, ενεργοποιεί την εκπαίδευση μισής ακρίβειας, η οποία εξοικονομεί χώρο χρησιμοποιώντας 16 bit αντί για 32 για την αποθήκευση των βαρών του μοντέλου. Διαθέσιμο μόνο όταν χρησιμοποιείται CUDA/GPU.
- ✓ **eval\_ratio**: Η αναλογία των δεδομένων που παρέχονται στο `input_filepath` που θα χρησιμοποιηθούν για την αξιολόγηση. Εάν παρέχεται `eval_filepath`, αυτό το όρισμα αγνοείται και το `input_filepath` χρησιμοποιείται μόνο ως δεδομένα εκπαίδευσης.

- ✓ `save_steps`: Αναλογία του συνολικού βήματος της εκπαίδευσης πριν από την αποθήκευση.
- ✓ `eval_steps`: Αναλογία του συνολικού βήματος της εκπαίδευσης πριν από την αξιολόγηση.
- ✓ `logging_steps`: Αναλογία του συνολικού βήματος της εκπαίδευσης πριν από την καταγραφή.
- ✓ `output_dir`: Ένας κατάλογος εξόδου στον οποίο θα αποθηκευτούν τα μοντέλα, εάν είναι ενεργοποιημένο το `save_steps`. Στο μέλλον ενδέχεται να προστεθούν και άλλα χαρακτηριστικά που αξιοποιούν αυτόν τον κατάλογο.

Parameter	Default
<code>learning_rate</code>	5e-5
<code>num_train_epochs</code>	1
<code>batch_size</code>	1
<code>preprocessing_processes</code>	1
<code>save_path</code>	""
<code>load_path</code>	""
<code>max_input_length</code>	None
<code>max_output_length</code>	None
<code>fp16</code>	False
<code>eval_ratio</code>	0.1
<code>save_steps</code>	0.0
<code>eval_steps</code>	0.1
<code>logging_steps</code>	0.1
<code>output_dir</code>	"happy_transformer"

Εικόνα 3.7: Παράμετροι που περιέχονται στην κλάση δεδομένων `TTTrainArgs` με τις `default` τιμές τους

Το μοντέλο T5 διατίθεται σε διάφορα μεγέθη: το small, το base, το large και τα 3B και 11B. Σε αυτή την εργασία θα χρησιμοποιηθεί το t5-base, το οποίο θεωρείται το βασικό μοντέλο για τέτοιου είδους επεξεργασίες.

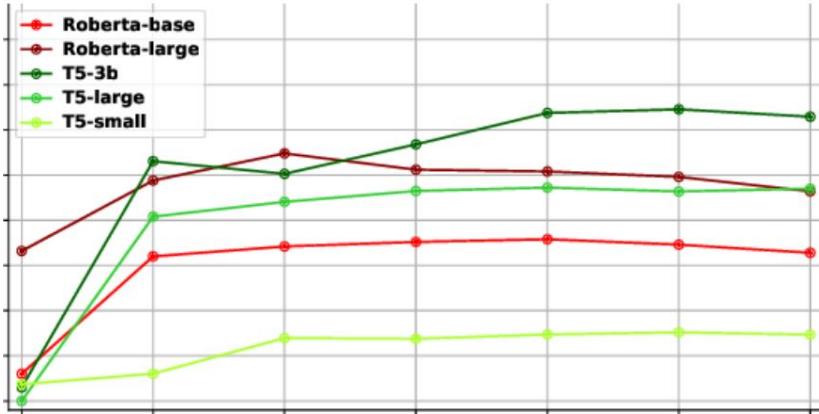
- ✓ Base [25]. Αυτό είναι το βασικό μας μοντέλο. Έχει περίπου 220 εκατομμύρια παραμέτρους.
- ✓ Small[26]. Είναι ένα μικρότερο μοντέλο με μόνο 6 στρώματα στον κωδικοποιητή και τον αποκωδικοποιητή. Αυτή η παραλλαγή έχει περίπου 60 εκατομμύρια παραμέτρους.
- ✓ Large [27]. Αυτή η παραλλαγή χρησιμοποιεί 24 στρώματα στον κωδικοποιητή και τον αποκωδικοποιητή, με αποτέλεσμα περίπου 770 εκατομμύρια παραμέτρους.
- ✓ 3B και 11B. Για την παραλλαγή "3B", χρησιμοποιούμε έναν κωδικοποιητή και αποκωδικοποιητή 24 επιπέδων, κάτι το οποίο οδηγεί σε περίπου 2,8 δισεκατομμύρια παραμέτρους [28] ενώ για την παραλλαγή "11B" χρησιμοποιούμε έναν κωδικοποιητή και αποκωδικοποιητή 24 επιπέδων παράγοντας ένα μοντέλο με περίπου 11 δισεκατομμύρια παραμέτρους [29].

## Model size variants

Model	Parameters	# layers	$d_{model}$	$d_{ff}$	$d_{kv}$	# heads
Small	60M	6	512	2048	64	8
Base	220M	12	768	3072	64	12
Large	770M	24	1024	4096	64	16
3B	3B	24	1024	16384	128	32
11B	11B	24	1024	65536	128	128

Εικόνα 3.8: Παραλλαγές του μοντέλου T5 με τις αντίστοιχες πληροφορίες

Παρακάτω παρουσιάζεται η απόδοση των μοντέλων T5 σε σχέση και με άλλα μοντέλα [30], όπου παρατηρούμε πως το T5-3b έχει καλύτερα αποτελέσματα.



Εικόνα 3.9: Αποτελέσματα αξιολόγησης των πέντε μοντέλων με διαφορετικά μεγέθη δεδομένων, που αντιστοιχούν σε 0, 1, 5, 10, 33, 50 και 100% των συνθετικών δεδομένων. Κάθε σημείο αντιπροσωπεύει τη μέση απόδοση ενός μοντέλου για τα πέντε benchmark.

Έτσι λοιπόν χρησιμοποιώντας την κλάση HappyTextToText, θέτουμε ως παραμέτρους αρχικά τον τύπο του μοντέλου που θα χρησιμοποιήσουμε με κεφαλαία γράμματα (T5) και έπειτα το όνομα του μοντέλου (t5-base).

```
happy_tt = HappyTextToText("T5", "t5-base")
```

Εικόνα 3.10: Ορισμός τύπου και ονόματος μοντέλου που θα χρησιμοποιηθεί

Χρησιμοποιούμε την παραλλαγή t5-base του μοντέλου T5 καθώς παρουσιάζει σημαντικά πλεονεκτήματα:

- ✓ Ισορροπία μεταξύ επιδόσεων και υπολογιστικού κόστους: Η παραλλαγή t5-base προσφέρει μια καλή ισορροπία μεταξύ των επιδόσεων του μοντέλου και των απαιτούμενων υπολογιστικών πόρων (μνήμη, χρόνος υπολογισμού κ.λπ.). Συνήθως παρέχει καλύτερες επιδόσεις από το t5-small λόγω του μεγαλύτερου μεγέθους του και των περισσότερων παραμέτρων, γεγονός που του επιτρέπει να καταγράφει πιο σύνθετα πρότυπα στα δεδομένα. Ενώ το t5-large και το t5-3b μπορεί να προσφέρουν καλύτερες επιδόσεις, είναι σημαντικά πιο απαιτητικά σε πόρους, απαιτώντας περισσότερη μνήμη και μεγαλύτερους χρόνους εκπαίδευσης, κάτι που μπορεί να μην είναι εφικτό για όλα τα περιβάλλοντα ή τις περιπτώσεις χρήσης.
- ✓ Δυναμικότητα μοντέλου: Το t5-base έχει 220 εκατομμύρια παραμέτρους, το οποίο παρέχει επαρκή χωρητικότητα για να μαθαίνει από μεγάλα σύνολα δεδομένων και να καταγράφει τις διαφοροποιήσεις που απαιτούνται για την αποτελεσματική διόρθωση γραμματικών λαθών χωρίς να είναι υπερβολικά απαιτητικό σε πόρους.

- ✓ Υιοθέτηση από την κοινότητα και Benchmarking: Η παραλλαγή t5-base χρησιμοποιείται ευρέως στην ερευνητική κοινότητα και χρησιμοποιείται συχνά ως βάση για διάφορες εργασίες, συμπεριλαμβανομένης της GEC. Αυτό διευκολύνει τη σύγκριση των αποτελεσμάτων με άλλες μελέτες και την αξιοποίηση της υπάρχουσας έρευνας και των βελτιστοποιήσεων.
- ✓ Ευελιξία: Τα μοντέλα T5 έχουν σχεδιαστεί για να χειρίζονται μια ποικιλία εργασιών NLP με ενιαία αρχιτεκτονική. Το t5-base είναι μια ευέλικτη επιλογή που μπορεί να ρυθμιστεί λεπτομερώς για συγκεκριμένες εργασίες, όπως η GEC, ενώ παράλληλα μπορεί να προσαρμοστεί και σε άλλες εργασίες αν χρειαστεί.
- ✓ Περιορισμοί υλικού: Η επιλογή του μοντέλου συχνά περιορίζεται από το διαθέσιμο υλικό. Το t5-base είναι πιο εφικτό να εκτελεστεί σε κοινές ρυθμίσεις όπως μεμονωμένες GPU ή mid-range cloud instances σε σύγκριση με μεγαλύτερα μοντέλα.
- ✓ Μέγεθος και ποιότητα συνόλου δεδομένων: Για μικρότερα σύνολα δεδομένων, τα μεγαλύτερα μοντέλα μπορεί να υπερπροσαρμόζονται, ενώ τα μικρότερα μοντέλα όπως το t5-small μπορεί να υπολειτουργούν. Το t5-base προσφέρει μια μέση λύση που λειτουργεί καλά για πολλά σύνολα δεδομένων.

Πολλά dataset έχουν χρησιμοποιηθεί κατά καιρούς για τη διόρθωση γραμματικών λαθών, όχι μόνο στην Αγγλική γλώσσα.

Language	Corpus	Sentences	Err. r.
English	Lang-8	1 147 451	14.1%
	NUCLE	57 151	6.6%
	FCE	33 236	11.5%
	W&I+LOCNESS	43 169	11.8%
Czech	AKCES-GEC	42 210	21.4%
German	Falko-MERLIN	24 077	16.8%
Russian	RULEC-GEC	12 480	6.4%

Εικόνα 3.11: Στατιστικά στοιχεία των διαθέσιμων σωμάτων δεδομένων για τη διόρθωση γραμματικών λαθών σε άλλες γλώσσες πέραν της Αγγλικής

Το πιο ευρέως χρησιμοποιούμενο σύνολο δεδομένων στη GEC είναι το NUCLE, το οποίο έχει απλή απόδοση προτάσεων, θεματική ενότητα και L1. Για αυτό, χρειαζόμαστε ένα διαφορετικό dataset που να περιέχει ποικιλία λαθών για καλύτερη εκπαίδευση.

Level	Type	Number	%
1	PUNCT	824.5	63.28
	ORTH	478.5	36.72
	<b>Total</b>	<b>1303.0</b>	<b>55.45</b>
2	SPELL	0.5	0.14
	PUNCT	229.5	63.31
	PREP	1.0	0.28
	OTHER	124.5	34.34
	NOUN:POSS	3.5	0.97
	NOUN	2.0	0.55
	DET	0.5	0.14
	ADJ	1.0	0.28
3	<b>Total</b>	<b>362.5</b>	<b>15.43</b>
	WO	9.5	1.39
	VERB:TENSE	37.5	5.48
	VERB:SVA	19.0	2.78
	VERB:INFL	1.0	0.15
	VERB:FORM	37.5	5.48
	VERB	40.0	5.84
	SPELL	115.5	16.87
	SPACE	11.0	1.61
	PRON	34.0	4.97
	PREP	69.0	10.08
	PART	4.0	0.58
	OTHER	110.0	16.07
	NOUN:POSS	3.5	0.51
	NOUN:NUM	35.5	5.19
	NOUN:INFL	2.5	0.37
	NOUN	35.5	5.19
	MORPH	28.0	4.09
	DET	57.0	8.33
	CONTR	4.0	0.58
	CONJ	3.5	0.51
	ADV	15.0	2.19
	ADJ:FORM	2.5	0.37
	ADJ	9.5	1.39
	<b>Total</b>	<b>684.5</b>	<b>29.13</b>

Εικόνα 3.12: Κατανομή τύπων σφαλμάτων σε διαφορετικό από το NUCLE dataset (ErAConD)

### 3.3 Επιλογή Dataset

Για την εκπαίδευση του μοντέλου χρησιμοποιείται ένα JFLEG σύνολο δεδομένων. Το JFLEG (JHU FLuency-Extended GUG) είναι ένα σώμα δεδομένων για τη διόρθωση γραμματικών λαθών (GEC) στα αγγλικά. Αποτελεί ένα χρυσό πρότυπο αναφοράς για την ανάπτυξη και την αξιολόγηση συστημάτων GEC όσον αφορά την ευχέρεια (κατά πόσο ένα κείμενο μοιάζει με τη μητρική του γλώσσα) καθώς και τη γραμματική [31]. Για κάθε αρχικό κείμενο (sentence), υπάρχουν τέσσερις διορθώσεις γραμμένες από παρατηρητές (corrections) ακολουθώντας συγκεκριμένες οδηγίες [32].

Please correct the following sentence to make it sound natural and fluent to a native speaker of (American) English. The sentence is written by a second language learner of English. You should fix grammatical mistakes, awkward phrases, spelling errors, etc. following standard written usage conventions, but your edits must be conservative. Please keep the original sentence (words, phrases, and structure) as much as possible. The ultimate goal of this task is to make the given sentence sound natural to native speakers of English without making unnecessary changes. Please do not split the original sentence into two or more. Edits are not required when the sentence is already grammatical and sounds natural.

Εικόνα 3.13: Οδηγίες σχολιασμού JFLEG

So I think we can not live if old people could not find sciences and technologies and they did not developed .	[ "So I think we would not be alive if our ancestors did not develop sciences and technologies . ", "So I think we could not live if older people did not develop science and technologies . ", "So I think we can not live if old people could not find science and technologies and they did not develop . ", "So I think we can not live if old people can not find the science and technology that has not been developed . " ]
For not use car .	[ "Not for use with a car . ", "Do not use in the car . ", "Car not for use . ", "Can not use the car . " ]
Here was no promise of morning except that we looked up through the trees we saw how low the forest had swung .	[ "Here was no promise of morning , except that we looked up through the trees , and we saw how low the forest had swung . "... ]
Thus even today sex is considered as the least important topic in many parts of India .	[ "Thus , even today , sex is considered as the least important topic in may parts of India . ", "Thus , even today , sex is..." ]

Εικόνα 3.14: Στιγμιότυπο του JFLEG dataset με την αρχική πρόταση (αριστερά) και τις αντίστοιχες διορθώσεις (δεξιά)

Το dataset περιέχει στο σύνολο 1.503 εγγραφές και είναι χωρισμένο σε δύο περίπου ισόποσα τμήματα το validation (755 γραμμές) και το test (748 γραμμές). Το σύνολο δεδομένων είναι διαθέσιμο στο δίκτυο παροχής συνόλων δεδομένων του Hugging Face και μπορούμε να έχουμε πρόσβαση χρησιμοποιώντας τη βιβλιοθήκη του Datasets. Δεν χρειάζεται να την εγκαταστήσουμε και μπορούμε να προχωρήσουμε κατευθείαν στην εισαγωγή μιας συνάρτησης που ονομάζεται `load_dataset` από τη βιβλιοθήκη.

```
from datasets import load_dataset

train_dataset = load_dataset("jfleg", split='validation[:]')
eval_dataset = load_dataset("jfleg", split='test[:]')
```

Εικόνα 3.15: Συλλογή του σώματος δεδομένων

Όπως φαίνεται παραπάνω, ως πρώτη παράμετρος χρησιμοποιείται το `id` του dataset (`jfleg`) ώστε να γίνει η σωστή επιλογή δεδομένων ενώ ως δεύτερη παράμετρος χρησιμοποιείται το αντίστοιχο τμήμα του συνόλου δεδομένων (`validation` ή `test`). Το σύνολο `validation` θα χρησιμοποιηθεί για εκπαίδευση (`train_dataset`) του μοντέλου και το σύνολο `test` για την αξιολόγησή του (`eval_dataset`).

### 3.4 Προεπεξεργασία Δεδομένων

Πριν προχωρήσουμε στην εκπαίδευση του μοντέλου χρειάζεται να γίνει προεπεξεργασία των δεδομένων μας ώστε να διορθωθούν τυχόν λάθη που περιέχουν και να βελτιώσουμε την απόδοση της γραμματικής διόρθωσης. Ορισμένες από τις περιπτώσεις εντός του συνόλου δεδομένων περιέχουν υπερβολικά πολλά κενά και εάν δεν διορθωθούν, το μοντέλο θα παράγει κενά όταν δεν απαιτείται. Για αυτό εφαρμόζεται ο παρακάτω κώδικας για να διορθώσουμε το κείμενο εισόδου και εξόδου τόσο για τα δεδομένα εκπαίδευσης όσο και για τα δεδομένα αξιολόγησης. Ο πίνακας replacements περιέχει σημεία στίξης, αριθμούς και γράμματα που ακολουθούνται από υπερβολικά κενά καθώς επίσης και τις αντίστοιχες διορθώσεις τους έχοντας αφαιρέσει αυτά τα κενά. Έπειτα, χρησιμοποιείται μια συνάρτηση remove\_spaces που δέχεται ως όρισμα κάποια text, τα οποία θα είναι οι αρχικές προτάσεις (sentences) και οι διορθώσεις τους (corrections) όπως θα δούμε παρακάτω. Η λειτουργία αυτής της συνάρτησης είναι να δέχεται το κείμενο που της δίνεται ως όρισμα και έπειτα να διατρέχει τον πίνακα replacements ώστε να εντοπίσει αν στην είσοδο υπάρχει κάποιο από τα υπάρχοντα λάθη. Αν τυχόν εντοπιστεί κάποιο από αυτά, γίνεται η αντικατάστασή του με την αντίστοιχη διόρθωσή του και η συνάρτηση επιστρέφει πλέον το διορθωμένο κείμενο.

```
replacements = [
    (" .", "."),
    (" ,", ","),
    (" '", "'"),
    (" ?", "?"),
    (" !", "!="),
    (" :", "!="),
    (" ;", "!="),
    (" n't", "n't"),
    ("2 0 0 6", "2006"),
    ("5 5", "55"),
    ("4 0 0", "400"),
    ("1 7-5 0", "1750"),
    ("2 0 %", "20%"),
    ("5 0", "50"),
    ("1 2", "12"),
    ("1 0", "10"),
]

def remove_spaces(text):
    for rep in replacements:
        text = text.replace(rep[0], rep[1])
    return text
```

Εικόνα 3.16: Συνάρτηση διόρθωσης υπερβολικών κενών

Επόμενο κομμάτι της προεπεξεργασίας είναι να φέρουμε το dataset στην κατάλληλη μορφή για το Happy Transformer. Πρέπει να δομήσουμε τόσο τα δεδομένα εκπαίδευσης όσο και τα δεδομένα αξιολόγησης στην ίδια μορφή, η οποία είναι ένα αρχείο CSV με δύο στήλες: είσοδος (input) και στόχος (target). Η στήλη εισόδου περιέχει γραμματικά λανθασμένο κείμενο και η στήλη στόχου περιέχει κείμενο που είναι η διορθωμένη έκδοση του κειμένου από τη στήλη εισόδου.

input	target
grammar: I has poor grammars	I have poor grammar
grammar: I wants too plays	I want to play

Εικόνα 3.17: Παράδειγμα μορφής αρχείου CSV με τις δύο στήλες input και target.

Παρακάτω παρατίθεται κώδικας που επεξεργάζεται τα δεδομένα στη σωστή μορφή. Χρησιμοποιείται μια συνάρτηση generate\_csv για τη δημιουργία των CSV αρχείων όπου δέχεται δύο ορίσματα. Αυτά είναι το όνομα του αρχείου CSV που θέλουμε να δημιουργήσουμε (csv\_path) και το σύνολο δεδομένων από το οποίο θα προκύψει αυτό το αρχείο (dataset). Ανοίγοντας το αρχείο δημιουργούμε τις επικεφαλίδες input και target ώστε να μπει η κάθε πρόταση στη σωστή στήλη. Πλέον, κάθε λανθασμένη πρόταση πρέπει να αντιστοιχείται σε μια μοναδική διόρθωση κάθε φορά. Έτσι, όλες οι λανθασμένες προτάσεις θα εμφανίζονται 4 φορές στη στήλη input, μία φορά για καθεμία από τις 4 διορθώσεις που υπάρχουν στη στήλη target. Συνολικά δηλαδή, δημιουργούμε 3016 παραδείγματα εκπαίδευσης και 2988 παραδείγματα αξιολόγησης.

```
input,target
grammar: New and new technology has been introduced to the society.,New technology has been introduced to society.
grammar: New and new technology has been introduced to the society.,New technology has been introduced into the society.
grammar: New and new technology has been introduced to the society.,Newer and newer technology has been introduced into society.
grammar: New and new technology has been introduced to the society.,Newer and newer technology has been introduced to the society.
```

Εικόνα 3.18: Παράδειγμα αντιστοίχισης καθεμίας διόρθωσης με την λανθασμένη πρόταση είσοδο

Συνεχίζοντας, πρέπει να καθορίσουμε την εργασία που θέλουμε να εκτελέσουμε προσθέτοντας το ίδιο πρόθεμα σε κάθε είσοδο. Σε αυτή την περίπτωση, θα χρησιμοποιήσουμε το πρόθεμα "grammar:". Αυτό γίνεται επειδή τα μοντέλα T5 είναι σε θέση να εκτελούν πολλαπλές εργασίες, όπως μετάφραση και περίληψη, με ένα μόνο μοντέλο και ένα μοναδικό πρόθεμα να χρησιμοποιείται για κάθε εργασία, ώστε το μοντέλο να μαθαίνει ποια εργασία πρέπει να εκτελέσει. Πρέπει επίσης να παραλείψουμε τις περιπτώσεις που περιέχουν κενό αλφαριθμητικό για να αποφύγουμε σφάλματα κατά τη λεπτομερή ρύθμιση. Έτσι για κάθε είσοδο προσθέτουμε στην αρχή της το πρόθεμα για διόρθωση γραμματικών λαθών και ελέγχουμε αν υπάρχει κάποιο από τα λάθη που αναφέρθηκαν παραπάνω, κάτι το οποίο γίνεται και για κάθε πρόταση στόχο. Έπειτα από αυτά, καθεμία γραμμή προστίθεται στο αρχείο CSV έως ότου προστεθούν όλες και ολοκληρωθεί η διαδικασία.

```
import csv

def generate_csv(csv_path, dataset):
    with open(csv_path, 'w', newline='') as csvfile:
        our_csv = csv.writer(csvfile)
        our_csv.writerow(["input", "target"])
        for sent in dataset:
            # Adding the task's prefix to input
            input_text = "grammar: " + sent["sentence"]
            for correction in sent["corrections"]:
                # a few of the cases contain blank strings.
                if input_text and correction:
                    input_text = remove_spaces(input_text)
                    correction = remove_spaces(correction)
                    our_csv.writerow([input_text, correction])

generate_csv("train.csv", train_dataset)
generate_csv("eval.csv", eval_dataset)
```

Εικόνα 3.19: Συνάρτηση δημιουργίας CSV αρχείων για εκπαίδευση (train.csv) και αξιολόγηση (eval.csv) του μοντέλου.

### 3.5 Εκπαίδευση Μοντέλου

Αφού έχουμε τελειώσει όλη τη διαδικασία της προεπεξεργασίας είμαστε έτοιμοι να εκπαιδεύσουμε το μοντέλο μας. Θα χρησιμοποιήσουμε τη μέθοδο `train()` που περιέχει η κλάση `HappyTextToText` και είναι ιδανική για την λεπτομερή ρύθμιση του μοντέλου για τη μετατροπή ενός αυτόνομου κειμένου σε ένα άλλο αυτόνομο κείμενο.

Πριν από αυτό, χρησιμοποιούμε την κλάση `TTTrainArgs` για να θέσουμε τις παραμέτρους της εκπαίδευσης. Για λόγους απλότητας, θα χρησιμοποιήσουμε τις προεπιλεγμένες παραμέτρους, εκτός από το μέγεθος της ομαδοποίησης (`batch_size`), το οποίο θα αυξήσουμε σε 8. Αυτό μας επιτρέπει να επιταχύνουμε τη διαδικασία ώστε να εξάγουμε πιο γρήγορα τα αποτελέσματά μας.

Η μέθοδος `train()` δέχεται τρεις εισόδους οι οποίες είναι οι παρακάτω:

- ✓ `input_filepath (string)`: ένα μονοπάτι προς ένα αρχείο CSV όπως φαίνεται στην Εικόνα 32.
- ✓ `args (TTTrainArgs)`: μια κλάση δεδομένων με τους ίδιους τύπους πεδίων όπως φαίνεται στην Εικόνα 25.
- ✓ `eval_filepath (string)`: By default θα δημιουργηθεί ένα σύνολο δεδομένων αξιολόγησης από τα παρεχόμενα δεδομένα εκπαίδευσης. Ωστόσο, μπορούμε να δώσουμε ένα `filepath` σε ένα αρχείο CSV όπως περιγράφεται για το `input_filepath` για να χρησιμοποιήσουμε αυτόνομα δεδομένα αξιολόγησης.

Στη συγκεκριμένη περίπτωση η μέθοδος δέχεται δύο ορίσματα. Το πρώτο είναι το όνομα του αρχείου πάνω στο οποίο θέλουμε να γίνει η εκπαίδευση (`train.csv`) και το δεύτερο είναι οι παράμετροι (`arguments`) σύμφωνα με τις οποίες επιθυμούμε να γίνει η εκπαίδευση. Το σύνολο δεδομένων αξιολόγησης θα δημιουργηθεί by default.

```
args = TTTrainArgs(batch_size=8)
happy_tt.train("train.csv", args=args)
```

Εικόνα 3.20: Εκπαίδευση μοντέλου

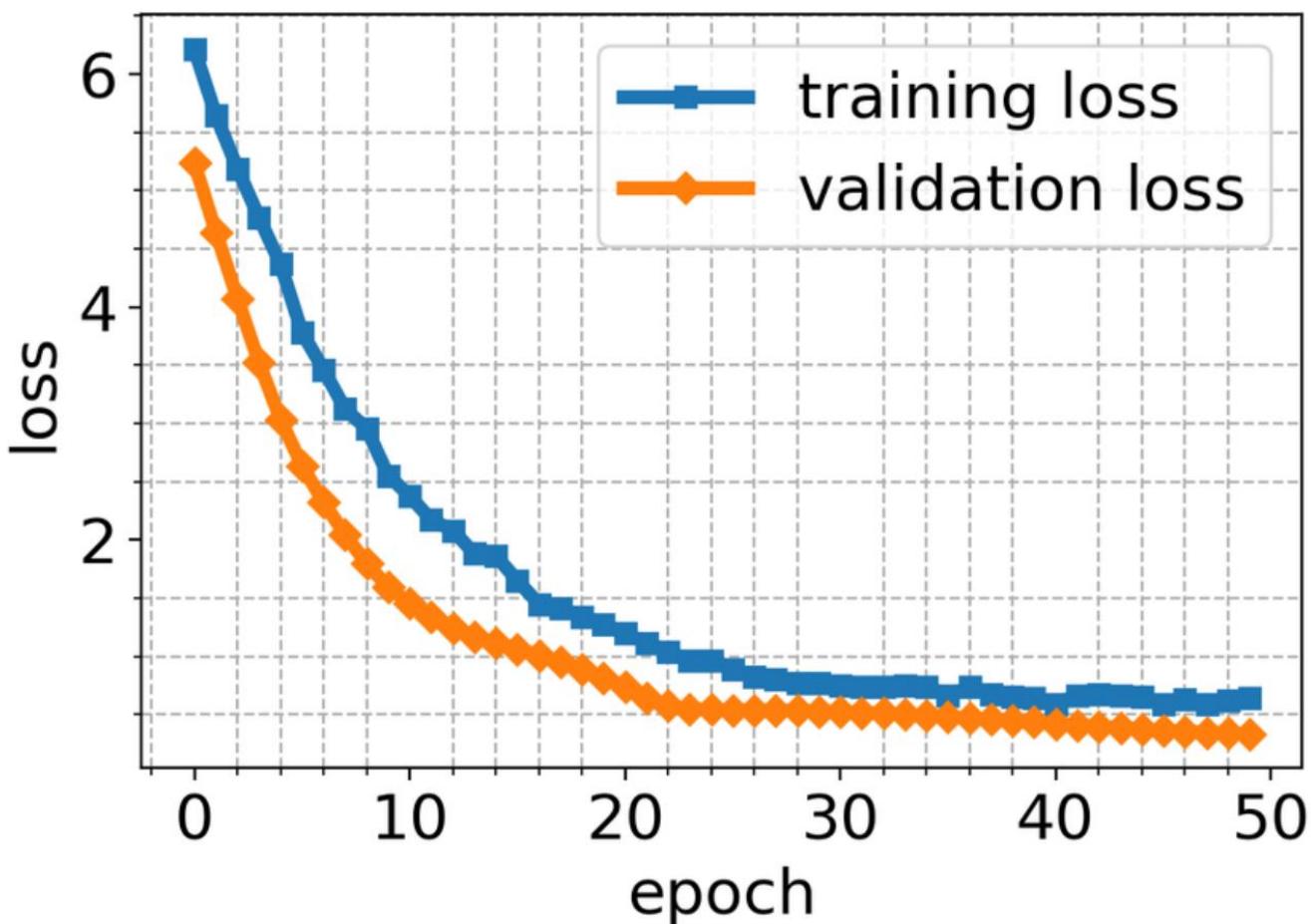
Τα αποτελέσματα που προκύπτουν από την εκπαίδευση είναι τα εξής:

Step	Training Loss	Validation Loss
1	1.321700	1.138514
34	0.834000	0.684559
68	0.751300	0.575330
102	0.673600	0.542197
136	0.635700	0.522847
170	0.639400	0.512090
204	0.612600	0.502808
238	0.613800	0.498251
272	0.567500	0.495194
306	0.549800	0.493305
340	0.555600	0.492933

Εικόνα 3.21: Αποτελέσματα μετά την εκπαίδευση του μοντέλου

Στην παραπάνω εικόνα διακρίνονται δύο απώλειες του μοντέλου που προκύπτουν έπειτα από κάθε step, η απώλεια εκπαίδευσης (training loss) και η απώλεια εγκυρότητας (validation loss). Η απώλεια εκπαίδευσης είναι μια μετρική που χρησιμοποιείται για την αξιολόγηση του τρόπου με τον οποίο ένα μοντέλο ταιριάζει στα δεδομένα εκπαίδευσης [33]. Δηλαδή, αξιολογεί το σφάλμα του μοντέλου στο σύνολο εκπαίδευσης. Σημειώνεται ότι το σύνολο εκπαίδευσης είναι ένα τμήμα ενός συνόλου δεδομένων που χρησιμοποιείται για την αρχική εκπαίδευση του μοντέλου. Υπολογιστικά, η απώλεια εκπαίδευσης υπολογίζεται λαμβάνοντας το άθροισμα των σφαλμάτων για κάθε παράδειγμα στο σύνολο εκπαίδευσης. Είναι επίσης σημαντικό να σημειωθεί ότι η απώλεια εκπαίδευσης μετριέται μετά από κάθε παρτίδα. Αυτό συνήθως απεικονίζεται με τη χάραξη μιας καμπύλης της απώλειας εκπαίδευσης.

Η απώλεια εγκυρότητας είναι μια μετρική που χρησιμοποιείται για την αξιολόγηση της απόδοσης ενός μοντέλου στο σύνολο εγκυρότητας (validation set). Το σύνολο εγκυρότητας είναι ένα τμήμα του συνόλου δεδομένων που έχει οριστεί για την επικύρωση της απόδοσης του μοντέλου. Η απώλεια εγκυρότητας είναι παρόμοια με την απώλεια εκπαίδευσης και υπολογίζεται από το άθροισμα των σφαλμάτων για κάθε παράδειγμα στο σύνολο εγκυρότητας. Επιπλέον, η απώλεια εγκυρότητας μετριέται μετά από κάθε επανάληψη. Αυτό μας πληροφορεί για το αν το μοντέλο χρειάζεται περαιτέρω ρύθμιση ή προσαρμογές, ή όχι. Για να γίνει αυτό, συνήθως σχεδιάζουμε μια καμπύλη μάθησης για την απώλεια εγκυρότητας [34].



Εικόνα 3.22: Ενδεικτικές καμπύλες για *training* και *validation* losses

Όπως παρατηρούμε από τα αποτελέσματα της εκπαίδευσης, οι απώλειες εκπαίδευσης και εγκυρότητας ακολουθούν τη διαδρομή των καμπυλών στην παραπάνω εικόνα. Αυτό σημαίνει πως καθ' όλη τη διάρκεια της εκπαίδευσης το validation loss είναι μικρότερο από το training loss, με τελική τιμή περίπου την ίδια.

### 3.6 Δοκιμή Μοντέλου με Παραδείγματα

Για να τεστάρουμε τη λειτουργία του μοντέλου θα χρησιμοποιήσουμε κάποια παραδείγματα. Το μοντέλο t5-base-grammar-correction είναι ιδανικό για αυτή τη διαδικασία καθώς είναι μια πιο ειδική περίπτωση του γενικού μοντέλου και μας επιτρέπει να εξάγουμε αποτέλεσμα μέσα από μεμονωμένες προτάσεις [35]. Όπως και το γενικό μοντέλο έτσι και αυτό εκπαιδεύτηκε με το πακέτο Happy Transformer χρησιμοποιώντας ένα JFLEG σύνολο δεδομένων.

Χρησιμοποιείται ένας αλγόριθμος δημιουργίας κειμένου που ονομάζεται άπληστος (greedy). Αυτός ο αλγόριθμος απλά επιλέγει την πιο πιθανά επόμενη λέξη. Μια κλάση που ονομάζεται TTSettings χρησιμοποιείται για τον έλεγχο του χρησιμοποιούμενου αλγορίθμου και των ρυθμίσεών του. Περνάει στην παράμετρο args για τη μέθοδο HappyTextToText.generate\_text(). Τα περιεχόμενα της κλάσης TTSettings δικρίνονται παρακάτω.

Parameter	Default	Definition
min_length	10	Minimum number of generated tokens
max_length	50	Maximum number of generated tokens
do_sample	False	When True, picks words based on their conditional probability
early_stopping	False	When True, generation finishes if the EOS token is reached
num_beams	1	Number of steps for each search path
temperature	1.0	How sensitive the algorithm is to selecting low probability options
top_k	50	How many potential answers are considered when performing sampling
top_p	1.0	Min number of tokens are selected where their probabilities add up to top_p
no_repeat_ngram_size	0	The size of an n-gram that cannot occur more than once. (0=infinity)

Εικόνα 3.23: Παράμετροι TTSettings με default τιμές και την επεξήγησή τους

Χρησιμοποιούμε τις παραμέτρους αυτές αυξάνοντας τη num\_beams σε 5, δηλαδή αυξάνοντας τα βήματα για κάθε μονοπάτι αναζήτησης, και μειώνοντας τη min\_length σε 1, αυξάνοντας το εύρος των generated tokens.

```
from happytransformer import HappyTextToText, TTSettings

happy_tt = HappyTextToText("T5", "vennify/t5-base-grammar-correction")
args = TTSettings(num_beams=5, min_length=1)

Downloading (...)lve/main/config.json: 100% [1.42k/1.42k] [00:00<00:00, 44.7kB/s]
Downloading pytorch_model.bin: 100% [892M/892M] [00:09<00:00, 32.1MB/s]
Downloading (...)okenizer_config.json: 100% [1.92k/1.92k] [00:00<00:00, 98.8kB/s]
Downloading spiece.model: 100% [792k/792k] [00:00<00:00, 29.3MB/s]
Downloading (...)/main/tokenizer.json: 100% [1.39M/1.39M] [00:00<00:00, 6.37MB/s]
Downloading (...)cial_tokens_map.json: 100% [1.79k/1.79k] [00:00<00:00, 71.0kB/s]
```

Εικόνα 3.24: Ορισμός τύπου και ονόματος μοντέλου που θα χρησιμοποιηθεί καθώς επίσης και των αντίστοιχων παραμέτρων του. Στην εικόνα επιπλέον φαίνονται τα χαρακτηριστικά που φορτώνουν παράλληλα με το μοντέλο.

Όπως αναφέρθηκε και παραπάνω θα χρησιμοποιηθεί η μέθοδος generate\_text() για τη διόρθωση γραμματικών λαθών. Η μέθοδος αυτή περιέχει δύο ορίσματα:

- ✓ text (string): Το κείμενο για το μοντέλο. Στη γενική περίπτωση η μέθοδος θα προσπαθήσει να συνεχίσει το κείμενο με τη δημιουργία δικού της κειμένου με βάση τα συμφραζόμενα.
- ✓ args (GENSettings): Είναι παρόμοια κλάση με την TTSettings. Έτσι και αυτή περιέχει τις παραμέτρους με βάση τις οποίες θα πραγματοποιηθεί η δημιουργία του κειμένου.

Αφού ολοκληρωθεί η διαδικασία για την οποία χρησιμοποιήθηκε η μέθοδος, επιστρέφεται ένα αντικείμενο με ένα μόνο πεδίο που ονομάζεται text.

```

from happytransformer import HappyGeneration, GENSettings
#-----#
happy_gen = HappyGeneration() # default uses gpt2
args = GENSettings(max_length=15)
result = happy_gen.generate_text("artificial intelligence is ", args=args)
print(result) # GenerationResult(text='\xa0a new field of research that has been gaining momentum in recent years.')
print('')
print(result.text) # a new field of research that has been gaining momentum in recent years.

```

Using pad\_token, but it is not set yet.

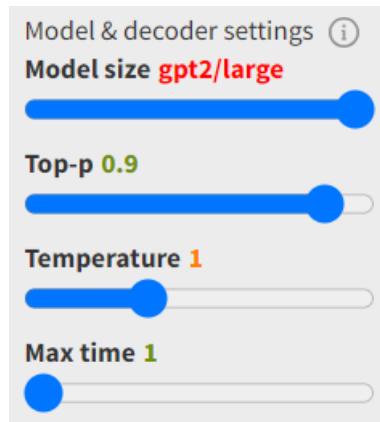
Setting `pad\_token\_id` to `eos\_token\_id`:50256 for open-end generation.

GenerationResult(text='\xa0a new field of research that has been gaining momentum in recent years.')

a new field of research that has been gaining momentum in recent years.

Εικόνα 3.25: Παράδειγμα λειτουργίας της μεθόδου generate\_text()

Στο συγκεκριμένο παράδειγμα επειδή η μέθοδος χρησιμοποιείται για παραγωγή κειμένου χρειάζεται η φόρτωση της κλάσης HappyGeneration η οποία χρησιμοποιεί by default το μοντέλο GPT-2 για την παραγωγή κειμένου. Το GPT-2 είναι ένα μοντέλο μετασχηματιστών που έχει προ-εκπαιδευτεί σε ένα πολύ μεγάλο σώμα αγγλικών δεδομένων με αυτοεπιβλεπόμενο τρόπο [36]. Αυτό σημαίνει ότι προ-εκπαιδεύτηκε μόνο στα ακατέργαστα κείμενα, χωρίς ανθρώπους να τα επισημάνουν με οποιονδήποτε τρόπο, με μια αυτόματη διαδικασία για τη δημιουργία εισόδων και επικετών από αυτά τα κείμενα. Γι' αυτό και μπορεί να χρησιμοποιήσει πολλά δημόσια διαθέσιμα δεδομένα. Πιο συγκεκριμένα, εκπαιδεύτηκε να μαντεύει την επόμενη λέξη στις προτάσεις. Το μοντέλο μαθαίνει μια εσωτερική αναπαράσταση της αγγλικής γλώσσας που μπορεί στη συνέχεια να χρησιμοποιηθεί για την εξαγωγή χαρακτηριστικών χρήσιμων για εργασίες, όπως η δική μας.



Εικόνα 3.26: Ρυθμίσεις μοντέλου και αποκωδικοποιητή

Generative Pre-trained Transformer 2 (GPT-2) is an open-source artificial intelligence created by OpenAI in February 2019.[1][2][3][4][5][6][7][8] GPT-2 translates text, answers questions, summarizes passages,[9] and performs other tasks.[10][11][12] It has been taught to speak English and Japanese , but not the rest of the world 's languages.[13] GPT-2 is designed with two goals in mind: To recognize patterns in sentences , and To understand complex sentences such as the ones above . It also uses language models that incorporate the principles of grammatical semantics.[14] The first goal is achieved with the use of a deep neural network . This network is built from an array of input modules and is trained by feeding the data into it.

Εικόνα 3.27: Συμπλήρωση GPT-2 χρησιμοποιώντας τον ιστότοπο Hugging Face Write With Transformer [37], με κείμενο από άρθρο της Wikipedia. Όλο το επισημασμένο κείμενο μετά την αρχική πρόταση έχει παραχθεί μηχανικά από την πρώτη προτεινόμενη συμπλήρωση, χωρίς περαιτέρω επεξεργασία

Ας δούμε λοιπόν τις δυνατότητες του μοντέλου και της μεθόδου μέσα από παραδείγματα. Θα δημιουργήσουμε κείμενα που περιέχουν πολλαπλά αλλά και μεμονωμένα γραμματικά λάθη, ώστε να μπορέσουμε να τα διορθώσουμε χρησιμοποιώντας το μοντέλο. Σε κάθε είσοδο πρέπει να προσθέτουμε το πρόθεμα grammar: στο κείμενο για να υποδείξουμε την εργασία που θέλουμε να εκτελέσουμε.

### 3.6.1 Παράδειγμα διόρθωσης ορθογραφίας σε κείμενο

- ✓ Διόρθωση ορθογραφίας σε κείμενο. Το μοντέλο αναγνωρίζει τα ορθογραφικά λάθη που περιέχονται στο κείμενο και τα διορθώνει μετατρέποντάς τα στη σωστή τους μορφή.

```
args = TTSettings(num_beams=5, min_length=1)
example_1 = "grammar: This sentence, has bad speling!"
result_1 = happy_tt.generate_text(example_1, args=args)
print(result_1.text)

This sentence, has bad spelling!
```

Εικόνα 3.28: Διόρθωση ορθογραφίας σε κείμενο

### 3.6.2 Παράδειγμα διόρθωσης γραμματικής σε κείμενο

- ✓ Διόρθωση γραμματικής σε κείμενο. Το μοντέλο αναγνωρίζει τα γραμματικά λάθη που περιέχονται στο κείμενο και τα διορθώνει μετατρέποντάς τα στη σωστή τους μορφή.

```
args = TTSettings(num_beams=5, min_length=1)
example_2 = "grammar: This sentence have bad grammar!"
result_2 = happy_tt.generate_text(example_2, args=args)
print(result_2.text)
```

```
This sentence has bad grammar!
```

Εικόνα 3.29: Διόρθωση γραμματικής σε κείμενο

### 3.6.3 Παράδειγμα διόρθωσης σημείων στίξης σε κείμενο

- ✓ Διόρθωση σημείων στίξης σε κείμενο. Το μοντέλο αναγνωρίζει τα λανθασμένα σημεία στίξης που περιέχονται στο κείμενο και με βάση τα συμφραζόμενα τα διορθώνει μετατρέποντάς τα στη σωστή τους μορφή.

```
args = TTSettings(num_beams=5, min_length=1)
example_3 = "grammar: What's wrong! This sentence has bad punctuation?"
result_3 = happy_tt.generate_text(example_3, args=args)
print(result_3.text)
```

```
What's wrong? This sentence has bad punctuation.
```

Εικόνα 3.30: Διόρθωση σημείων στίξης σε κείμενο

### 3.6.4 Παράδειγμα διόρθωσης πολλαπλών λαθών σε κείμενο

- ✓ Διόρθωση πολλαπλών λαθών σε κείμενο. Το μοντέλο αναγνωρίζει όλων των ειδών τα λάθη που περιέχονται στο κείμενο και με βάση τα συμφραζόμενα τα διορθώνει μετατρέποντάς τα στη σωστή τους μορφή.

```
args = TTSettings(num_beams=5, min_length=1)
example_4 = "grammar: Your you laptops is brokens. You wonted to practice on eror corection,"
result_4 = happy_tt.generate_text(example_4, args=args)
print(result_4.text)
```

Your laptop is broken. You won't practice on error corection.

Εικόνα 3.31: Διόρθωση πολλαπλών λαθών σε κείμενο

### 3.6.5 Παράδειγμα αναγνώρισης σωστού κειμένου

- ✓ Αναγνώριση σωστού κειμένου. Το μοντέλο αναγνωρίζει ότι το κείμενο είναι σωστό και δεν πραγματοποιεί καμία ενέργεια. Η έξοδός του είναι ίδια με την είσοδο.

```
args = TTSettings(num_beams=5, min_length=1)
example_5 = "grammar: This sentence is correct!"
result_5 = happy_tt.generate_text(example_5, args=args)
print(result_5.text)
```

This sentence is correct!

Εικόνα 3.32: Αναγνώριση σωστού κειμένου

Παρακάτω παρουσιάζεται ένα διάγραμμα ροής για τη μεθοδολογία εκπαίδευσης του μοντέλου, με όλα τα βήματα που ακολουθήθηκαν.



Εικόνα 3.33: Διάγραμμα ροής της μεθοδολογίας που ακολουθήθηκε

## 4. Χρήση μοντέλου σε ελληνικό dataset

Ένα μοντέλο T5 μπορεί να προσαρμοστεί για να λειτουργήσει σε ένα σύνολο ελληνικών δεδομένων για τη διόρθωση γραμματικών λαθών (GEC). Ωστόσο, υπάρχουν διάφορες εκτιμήσεις και βήματα που πρέπει να κάνουμε για να διασφαλίσουμε ότι θα λειτουργήσει αποτελεσματικά.

### 4.1 Επιλογή προ-εκπαιδευμένου μοντέλου

Τα τυποποιημένα μοντέλα T5, όπως το t5-base, έχουν προ-εκπαιδευτεί σε αγγλικά δεδομένα. Για να το χρησιμοποιήσουμε για τα ελληνικά, θα χρειαστεί ένα μοντέλο που έχει προ-εκπαιδευτεί σε ελληνικά ή πολύγλωσσα δεδομένα. Μπορεί να χρησιμοποιηθεί μια πολύγλωσση παραλλαγή του T5, το mT5 (πολύγλωσσο T5), το οποίο έχει προ-εκπαιδευτεί σε δεδομένα από πολλές γλώσσες, συμπεριλαμβανομένων των ελληνικών [38]. Η πολύγλωσση παραλλαγή mT5, η οποία έχει επιτύχει κορυφαία αποτελέσματα σε διάφορες εργασίες NLP στην αγγλική γλώσσα, προ-εκπαιδεύτηκε σε ένα σύνολο δεδομένων που καλύπτει 101 γλώσσες, ακολουθώντας το παράδειγμα της γενικότερης λύσης της προ-εκπαίδευσης σε πολλές γλώσσες. Τόσο ο T5 όσο και ο MT5 χρησιμοποιούν έναν μετασχηματιστή Encoder Decoder ο οποίος προ-εκπαιδεύεται με masked language μοντελοποίηση, καλύπτοντας διαδοχικά διαστήματα των tokens εισόδου και στη συνέχεια προσπαθώντας να τα ανακατασκευάσει. Ο T5 προ-εκπαιδεύτηκε σε 750 GB αγγλόφωνου κειμένου που προήλθε από τον δημόσιο ιστό Common Crawl. Ο mT5 προ-εκπαιδεύτηκε σε δεδομένα από όλα τα 71 μηνιαία δεδομένα ιστού που έχει δημοσιεύσει μέχρι στιγμής το Common Crawl, τα οποία είναι περισσότερα από τα δεδομένα προέλευσης που χρησιμοποιήθηκαν από το T5. Σελίδες με λίγους χαρακτήρες και σελίδες που περιείχαν κακές λέξεις αποκλείστηκαν. Στη συνέχεια, οι σελίδες ομαδοποιήθηκαν στις 101 γλώσσες που εντοπίστηκαν αυτόματα. Η πιο συχνή γλώσσα ήταν τα αγγλικά, ακολουθούμενη από τα ρωσικά και τα ισπανικά. Τα ελληνικά βρίσκονταν στην 20ή θέση, με 43 δισεκατομμύρια tokens που εξήχθησαν από 42 εκατομμύρια σελίδες. Δεδομένου ότι τα ελληνικά είναι μια καλά υποστηριζόμενη γλώσσα στα δεδομένα που χρησιμοποιήθηκαν για την προ-εκπαίδευση του mT5, το μοντέλο αυτό είναι κατάλληλος υποψήφιος για να χρησιμοποιηθεί για μεταγενέστερες εργασίες NLP στην ελληνική γλώσσα.

### 4.2 Λεπτομερής ρύθμιση και επιλογή dataset

Ανεξάρτητα από το προ-εκπαιδευμένο μοντέλο που θα επιλέξουμε, θα χρειαστεί να συντονίσουμε το μοντέλο σε ένα σύνολο δεδομένων ελληνικών. Η λεπτομερής ρύθμιση περιλαμβάνει την εκπαίδευση του μοντέλου στο συγκεκριμένο έργο της διόρθωσης γραμματικών λαθών σε ελληνικές προτάσεις.

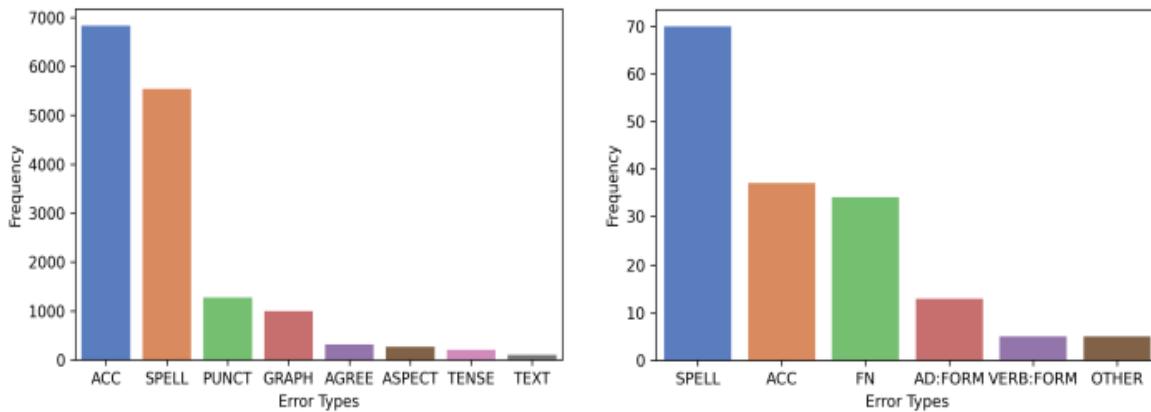
Σε πρόσφατες εργασίες έχουν χρησιμοποιηθεί δύο συγκεκριμένα σύνολα δεδομένων. Τα σύνολα δεδομένων που χρησιμοποιήθηκαν για τους σκοπούς της εργασίας ήταν το Greek Native Corpus και το Greek Learner Corpus, τα οποία ονομάστηκαν GNC και GLC αντίστοιχα. Το πρώτο είναι μια συλλογή δοκιμών (358 προτάσεις) γραμμένων από Έλληνες μαθητές γυμνασίου, τα οποία ψηφιοποιήθηκαν και σχολιάστηκαν, παρέχοντας τόσο διορθώσεις όσο και τύπους λαθών, ακολουθώντας μια ελληνική προσαρμογή του σχήματος σχολιασμού ERRANT. Παρά το γεγονός ότι το σύνολο δεδομένων του GNC μπορεί να χρησιμοποιηθεί για σκοπούς GEC, καμία εργασία στη βιβλιογραφία δεν έχει αναφέρει αντίστοιχα αποτελέσματα. Το GLC είναι μια συλλογή δοκιμών που γράφτηκαν από μαθητές της ελληνικής ως δεύτερης γλώσσας (μαθητές της GSL).

Original	Corrected
<p>Μια φορά κι έναν καιρό ήταν <b>τρεια</b> πουλιά και έτσι <b>όπος</b> έφεβγε η μαμά πλησίασε μια γάτα μετά τους χοιτούσε περίεργα</p>	<p>Μια φορά κι έναν καιρό ήταν <b>τρία</b> πουλιά και έτσι <b>όπως</b> έφευγε η μαμά πλησίασε μια γάτα μετά τους χοιτούσε περίεργα.</p>

Εικόνα 4.1: Παράδειγμα πρότασης GLC με διορθώσεις. Και τα τρία λάθη είναι ορθογραφικά λάθη. Η ευχέρεια της γραφής δεν είναι επίσης ικανοποιητική.

Δεδομένου ότι περιλαμβάνει μόνο τον σχολιασμό του τύπου λάθους και όχι τυχόν διορθώσεις, δεν μπορεί να χρησιμοποιηθεί για σκοπούς GEC. Ως εκ τούτου, και για να μπορέσει να χρησιμοποιηθεί το GLC ως σύνολο αξιολόγησης και για να είναι δυνητικά πιο ευέλικτο για σκοπούς GEC, αποφασίστηκε η διόρθωση των προτάσεων από έναν Έλληνα φιλόλογο, επεκτείνοντάς το στο Greek Learner Corpus Corrections ή GLC2.

Τα δύο σύνολα δεδομένων είναι τα πρώτα ελληνικά σύνολα δεδομένων που περιλαμβάνουν διορθώσεις που δημιουργούνται από ειδικούς, ενώ το GLC είναι επίσης το πρώτο πλήρες σύνολο δεδομένων με κείμενα που έχουν συγγράψει μαθητές της GSL. Αυτό είναι πολύ σημαντικό, καθώς οι περισσότερες τρέχουσες προσεγγίσεις NMT, όπως οι Encoder-Decoders, χρειάζονται μεγάλες ποσότητες δεδομένων (συμπεριλαμβανομένων των δεδομένων με ετικέτες σφαλμάτων) προκειμένου να είναι σε θέση να γενικεύσουν σωστά.



Εικόνα 4.2: Συχνότητα των συνηθέστερων τύπων σφαλμάτων στο GLC όπου καταδεικνύονται οι τύποι σφαλμάτων με συχνότητα πάνω από 100 εμφανίσεις (αριστερά) και συχνότητα των συνηθέστερων τύπων σφαλμάτων στο GNC, όπου παρουσιάζονται οι τύποι σφαλμάτων με συχνότητα πάνω από 5 εμφανίσεις (δεξιά).

### 4.3 Αξιολόγηση μοντέλου μέσω μετρικής ELERRANT

Για να αξιολογηθεί αυτό το μοντέλο έχει χρησιμοποιηθεί η μετρική ELERRANT [39], η οποία είναι μια παραλλαγή της ERRANT που χρησιμοποιείται για αγγλικά κείμενα. Προκειμένου να προσαρμοστεί η ERRANT στην ελληνική γλώσσα, χρησιμοποιήθηκε ο αρχικός ταξινομητής ERRANT ως πρότυπο. Αυτή η έκδοση χρησιμοποιεί το ελληνικό λεξικό ορθογραφικού ελέγχου Hunspell για τον εντοπισμό ορθογραφικών λαθών και το ελληνικό SpaCy ως κύριο POS tagger. Λόγω των μορφολογικών διαφορών μεταξύ των δύο γλωσσών (αγγλικά και ελληνικά), έχουν αφαιρεθεί ορισμένες κατηγορίες λαθών που υπάρχουν στον αρχικό ERRANT, ενώ προστέθηκαν κάποιες νέες.

Λόγω του γεγονότος ότι τα ελληνικά είναι μια γλώσσα με μεγάλη κλίση και οι περισσότεροι POS έχουν κάποιου είδους κλίση, συγχωνεύθηκαν κάποιοι τύποι σφαλμάτων προκειμένου να συμπεριληφθούν όσο το δυνατόν περισσότερες πληροφορίες σχετικά με το σφάλμα. Η απόφαση αυτή μπορεί να είναι θεωρηθεί ως συμβιβασμός, δεδομένου ότι πολλά λάθη μπορεί να έχουν περισσότερους από έναν επικαλυπτόμενους τύπους σφαλμάτων, επομένως με τη συγχώνευση των υποτύπων στο FORM διατηρείται η αμφισημία και η πολλαπλότητα του σφάλματος.

Error Type	Meaning	Description	Example
AD:FORM	Adverb Form	Errors concerning the form an adverb.	χαλός → χαλώς
ADJ:FORM*	Adjective Form	Errors concerning the form of an adjective	χαλός → χαλύτερος
NOUN:FORM	Noun Form	Errors concerning the number, the case or the suffix of a noun.	του νους → του νου
PRON:FORM	Pronoun Form	Errors concerning the number, the case or the suffix of a pronoun.	χάποια → χάποιας
VERB:FORM	Verb Form	Errors concerning the disposition, the voice, the inflection, the tense, the number or the person of a verb.	(εσείς) πηγαίνεται → (εσείς) πηγαίνετε
CONJ	Conjunction	Errors concerning conjunctions.	και → αλλά
PREP	Preposition	Errors concerning prepositions.	από → σε
DET*	Determiner	Errors concerning articles or determiners.	το → του τον → έναν
SPELL	Spelling	Spelling errors.	ευχέρια → ευχέρεια
FN	Final -v/nu	Final -v/nu addition or removal.	την → τη / μη → μην
PUNCT	Punctuation	Errors concerning the punctuation.	. → ;
OTHER	Other Errors	An error that does not fit into any other category but can still be corrected.	χαμία → για χανένα
ACC	Accentuation	Accentuation addition or removal.	χαθήκοντα → χαθήκοντα
UNK	Unknown error type	An error that can be detected but not corrected.	usually long error spans
WO	Words Order	Error in words order.	όταν φεύγω έρθεις → όταν έρθεις φεύγω
ORTH*	Orthography	Spacing Errors	γιασένα → για σένα
PART:FORM	Participle Form	Errors concerning the number, the case or the person of a participle.	(πήγε) τρεχόμενος → (πήγε) τρέχοντας
VERB:SVA	Subject Verb Agreement	The subject and the verb to be in person agreement.	(εγώ θα) φύγει → (εγώ θα) φύγω

Εικόνα 4.3: ELERRANT και οδηγός human error type annotation. Οι τύποι σφαλμάτων με τον αστερίσκο (\*) δεν υπάρχουν στον οδηγό human annotation, ενώ οι δύο τελευταίοι τύποι σφαλμάτων του πίνακα δεν υπάρχουν στον οδηγό σχολιασμού ELERRANT

Οι κυριότερες αλλαγές είναι οι εξής:

Προστέθηκε ο τύπος λάθους AD:FORM (Adverb Form), για να μεταφερθούν λάθη που αφορούν κυρίως τον συγκριτικό και υπερθετικό βαθμό του επιρρήματος. Η κατηγορία CONTR(Contraction) αφαιρέθηκε προσωρινά λόγω του γεγονότος ότι οι συμπτύξεις στα ελληνικά μπορούν να συμβούν σε κάθε λέξη που αρχίζει ή τελειώνει με φωνήν υπό ορισμένες προϋποθέσεις. Αυτή τη στιγμή αναπτύσσεται ένα λεξικό που συγκεντρώνει τις πιο συχνές περιπτώσεις συμπτύξεων στα ελληνικά και σχεδιάζεται να ενσωματωθεί σε μελλοντικές εκδόσεις. Οι λέξεις NOUN:INFL (Noun Inflection), NOUN:POSS (Noun Possessive) και NOUN:NUM (Noun Number) καταγράφονται στο NOUN:FORM (Noun Form). Τα PART (particles) επίσης απορρίφθηκαν, καθώς στην ελληνική γλώσσα έχουν διαφορετική λειτουργία, κυρίως στην κατασκευή χρονικών διαστημάτων. Προστέθηκε επίσης το PRON:FORM (Pronoun Form), καθώς οι αντωνυμίες είναι επίσης κλιτικές. Από τις κατηγορίες των ρημάτων, διατηρήθηκαν μόνο τα VERB, VERB:FORM και VERB:TENSE.

Προσθέθηκαν δύο ακόμη κατηγορίες από το μηδέν: τα ACC (Accent) και FN (Final - ν/ν). Ο τονισμός στα ελληνικά δηλώνεται με ένα σημάδι τονισμού (') και όχι απλώς με τον τονισμό της λέξης κατά την ομιλία. Η διατήρηση ή η παράλειψη του τελικού νυ σε ορισμένες ελληνικές λέξεις (άρθρα, αντωνυμίες ή μόρια), λόγω της συχνότητάς του ως λάθος ακόμη και από τους φυσικούς ομιλητές, θεωρείται διαφορετικό τύπου λάθος και όχι απλώς ορθογραφικό λάθος. Για τους δύο προαναφερθέντες τύπους λαθών, δημιουργήθηκαν δύο νέες αντίστοιχες συναρτήσεις οι οποίες προσθέθηκαν στο ELERRANT.

Label	Original Text	Corrected Text	Error Description	Error Type
e	Αρχικά, από την μεριά των μεγάλων καλό θα ήταν να κατανοήσουν πως ο υπερπροστατευτισμός αναστέλλει την υπευθυνότητα των νέων και προκαλεί συνήθως έντονες αντιδράσεις, αποτυγχάνοντας έτσι την προστασία τους και προκαλώντας μάλλον αντίθετα αποτελέσματα.	Αρχικά, από την μεριά των μεγάλων καλό θα ήταν να κατανοήσουν πως ο υπερπροστατευτισμός αναστέλλει την υπευθυνότητα των νέων και προκαλεί συνήθως έντονες αντιδράσεις, αποτυγχάνοντας έτσι την προστασία τους και προκαλώντας μάλλον αντίθετα αποτελέσματα.	R	FN
e	Αρχικά, από την μεριά των μεγάλων καλό θα ήταν να κατανοήσουν πως ο υπερπροστατευτισμός αναστέλλει την υπευθυνότητα των νέων και προκαλεί συνήθως έντονες αντιδράσεις, αποτυγχάνοντας έτσι την προστασία τους και προκαλώντας μάλλον αντίθετα αποτελέσματα.	Αρχικά, από τη μεριά των μεγάλων καλό θα ήταν να κατανοήσουν πως ο υπερπροστατευτισμός αναστέλλει την υπευθυνότητα των νέων και προκαλεί συνήθως έντονες αντιδράσεις, αποτυγχάνοντας έτσι την προστασία τους και προκαλώντας μάλλον αντίθετα αποτελέσματα.	R	SPELL
e	Αρχικά, από την μεριά των μεγάλων καλό θα ήταν να κατανοήσουν πως ο υπερπροστατευτισμός αναστέλλει την υπευθυνότητα των νέων και προκαλεί συνήθως έντονες αντιδράσεις, αποτυγχάνοντας έτσι την προστασία τους και προκαλώντας μάλλον αντίθετα αποτελέσματα.	Αρχικά, από τη μεριά των μεγάλων καλό θα ήταν να κατανοήσουν πως ο υπερπροστατευτισμός αναστέλλει την υπευθυνότητα των νέων και προκαλεί συνήθως έντονες αντιδράσεις, αποτυγχάνοντας έτσι την προστασία τους και προκαλώντας μάλλον αντίθετα αποτελέσματα.	R	PART:FORM

Εικόνα 4.4: Δείγμα σχολιασμού: τρεις καταχωρίσεις της ίδιας πρότασης με σχολιασμό κάθε φορά ενός διαφορετικού σφάλματος

#### 4.4 Χρήση και αξιολόγηση μοντέλου mT5 σε ελληνικό dataset

Όπως και για το μοντέλο T5 που χρησιμοποιήθηκε για τα κείμενα Αγγλικών, έτσι και για το μοντέλο mT5 για τα Ελληνικά ακολουθούμε την ίδια διαδικασία και τεχνική. Χρησιμοποιούμε δύο χειροκίνητα φτιαγμένα dataset, ένα για το fine-tuning του μοντέλου και ένα για την αξιολόγησή του, με 10 λανθασμένες προτάσεις και τις αντίστοιχες διορθώσεις τους. Τα dataset αυτά ακολουθούν τα πρότυπα τα οποία δέχεται ως είσοδο το μοντέλο mT5 και είναι επαρκή για τη σωστή του λειτουργία.

Sentence	Corrections
Έχω ένα βιβλίο .	Έχω ένα βιβλίο.
Αυτή είναι τι σπίτι μου .	Αυτή είναι το σπίτι μου.
Εμεις παμαι στο παρκο καθε μερα .	Εμείς πάμε στο πάρκο κάθε μέρα.
Αυτοι ειναι οι καλυτερη φιλη μου .	Αυτοί είναι οι καλύτεροι φίλοι μου.
Το σκυλος ειναι μεγαλο και γρηγορο .	Το σκυλί είναι μεγάλο και γρήγορο.

Εικόνα 4.5: Δείγμα περιεχομένων των δύο dataset (train & evaluation).

Ακολουθώντας λοιπόν τις ίδιες διαδικασίες, υπολογίζουμε την απώλεια και το GLEU score πριν το fine-tuning, κάνουμε fine-tuning το μοντέλο mT5 στο training dataset και έπειτα υπολογίζουμε τη νέα απώλεια και το αξιολογούμε με τη χρήση της μετρικής GLEU χρησιμοποιώντας το evaluation dataset. Τα αποτελέσματα που προκύπτουν μέσα από την αξιολόγηση του μοντέλου (που θα συζητηθεί πιο αναλυτικά στο κεφάλαιο 5) είναι τα παρακάτω.

Before loss: 1.7212671041488647 After loss: 1.3783318996429443

Εικόνα 4.6: Απώλεια πριν (αριστερά) και μετά (δεξιά) το fine-tuning.

GLEU Score: 0.0548 GLEU Score: 0.4093

Εικόνα 4.7: GLEU score πριν (αριστερά) και μετά (δεξιά) το fine-tuning.

Παρατηρώντας την απώλεια πριν και μετά το fine-tuning καταλαβαίνουμε πως αν και μειώνεται η απώλεια του μοντέλου, αυτή παραμένει σε πολύ υψηλά επίπεδα. Αυτό συμβαίνει λόγω του πολύ μικρού δείγματος δεδομένων που παρέχουμε στο μοντέλο, καθώς σε ένα πολύ μεγαλύτερο όγκο δεδομένων θα παρουσίαζε καλύτερα αποτελέσματα. Το ίδιο παρατηρούμε και στο GLEU score για το μοντέλο μας. Αντίστοιχα με πριν, το GLEU score αυξάνεται μετά το fine-tuning όμως παραμένει στα πολύ μικρά επίπεδα, περίπου στο 40%.

## 5. Αξιολόγηση Μοντέλου

### 5.1 Μετρικές Αξιολόγησης

Για την αξιολόγηση συστημάτων GEC έχουν χρησιμοποιηθεί διάφορες αυτόματες μετρικές. Το F-score, ο αρμονικός μέσος όρος της ακρίβειας (precision) και της ανάκλησης (recall), είναι μία από τις πιο συχνά χρησιμοποιούμενες μετρικές. Χρησιμοποιήθηκε ως επίσημη μετρική αξιολόγησης για διάφορες διαμοιρασμένες εργασίες, όπου οι συμμετέχοντες κλήθηκαν να εντοπίσουν και να διορθώσουν λάθη που αφορούν κλειστές κατηγορίες (για παράδειγμα προσδιορισμούς και εμπρόθετες προτάσεις) [40]. Παρόλα αυτά, υπάρχουν και άλλες διαδεδομένες μετρικές αξιολόγησης όπως η Max Match ( $M^2$ ) η οποία είναι η πιο συχνά χρησιμοποιούμενη μετρική αξιολόγησης στη GEC σήμερα, η I-Measure η οποία παρουσιάζει πάρα πολλά μειονεκτήματα, η BLEU (BiLateral Evaluation Understudy) η οποία έχει τον υψηλότερο βαθμό συσχέτισης με την ανθρώπινη αξιολόγηση και η ERRANT η οποία είναι μια βελτιωμένη έκδοση της μετρικής αξιολόγησης  $M^2$ .

Ας αναλύσουμε μερικές από τις βασικές μετρικές που χρησιμοποιούνται συνήθως για την αξιολόγηση των συστημάτων διόρθωσης γραμματικών λαθών (GEC). Αυτές οι μετρικές βοηθούν στην κατανόηση της απόδοσης και της αποτελεσματικότητας των μοντέλων GEC:

Ακρίβεια, ανάκληση και F-score:

**Ακρίβεια:** Αυτό μετρά το ποσοστό των διορθώσεων που πραγματοποιούνται από το σύστημα GEC και είναι πραγματικά σωστές. Η υψηλή ακρίβεια δείχνει ότι το μοντέλο κάνει λίγες λανθασμένες διορθώσεις.

**Ανάκληση:** Αυτό μετρά το ποσοστό των πραγματικών λαθών στο κείμενο που το σύστημα GEC εντοπίζει και διορθώνει σωστά. Η υψηλή ανάκληση δείχνει ότι το μοντέλο εντοπίζει τα περισσότερα λάθη.

**F-score:** Η F-score είναι ένας σταθμισμένος αρμονικός μέσος όρος της ακρίβειας και της ανάκλησης, δίνοντας μεγαλύτερη σημασία στην ακρίβεια. Είναι μια μετρική που χρησιμοποιείται συνήθως στο GEC για την εξισορρόπηση του συμβιβασμού μεταξύ ακρίβειας και ανάκλησης. Ένα υψηλό F-score υποδεικνύει ένα ισορροπημένο και αποτελεσματικό σύστημα GEC. Η ακρίβεια και η ανάκληση πρέπει να εξετάζονται μαζί. Ένα μοντέλο με υψηλή ακρίβεια αλλά χαμηλή ανάκληση μπορεί να μην διορθώνει αρκετά σφάλματα, ενώ ένα μοντέλο με υψηλή ανάκληση αλλά χαμηλή ακρίβεια μπορεί να εισάγει πάρα πολλές λανθασμένες διορθώσεις.

## GLEU:

Η GLEU είναι μια παραλλαγή της μετρικής BLEU, προσαρμοσμένη για την αξιολόγηση συστημάτων GEC. Μετρά την επικάλυψη n-γραμμάτων μεταξύ των διορθωμένων προτάσεων και των προτάσεων αναφοράς, λαμβάνοντας υπόψη τόσο την ακρίβεια όσο και την ανάκληση των n-γραμμάτων. Η GLEU παρέχει μια ισορροπημένη άποψη της απόδοσης του συστήματος GEC, λαμβάνοντας υπόψη τόσο την ευχέρεια όσο και την επάρκεια των διορθώσεων. Είναι ιδιαίτερα χρήσιμη για την καταγραφή λεπτών γραμματικών βελτιώσεων που μπορεί να διαφύγουν από άλλες μετρικές.

## I-measure:

Η I-measure είναι μια ολοκληρωμένη μετρική αξιολόγησης ειδικά σχεδιασμένη για την αξιολόγηση της απόδοσης των συστημάτων διόρθωσης γραμματικών λαθών (GEC). Σε αντίθεση με τις παραδοσιακές μετρικές όπως η ακρίβεια, η ανάκληση ή το F-score, οι οποίες εστιάζουν κυρίως στη δυαδική ταξινόμηση των σφαλμάτων (σωστά vs. λανθασμένα), η I-measure στοχεύει στην παροχή μιας πιο διαφοροποιημένης αξιολόγησης λαμβάνοντας υπόψη τη σοβαρότητα των σφαλμάτων και τη σημασία των διορθώσεων. Προσπαθεί να αντιμετωπίσει ορισμένες από τις αδυναμίες των παραδοσιακών μετρικών αξιολόγησης παρέχοντας μια πιο λεπτομερή και σταθμισμένη αξιολόγηση των διορθώσεων. Η ικανότητά της να λαμβάνει υπόψη τη σοβαρότητα και τον αντίκτυπο των σφαλμάτων την καθιστά πολύτιμο εργαλείο για την ανάπτυξη και την αξιολόγηση συστημάτων GEC που δίνουν προτεραιότητα σε ουσιαστικές και επιδραστικές διορθώσεις. Ωστόσο, η πολυπλοκότητα και η υποκειμενικότητα που συνδέονται με την εφαρμογή της αναδεικνύουν την ανάγκη προσεκτικής ταξινόμησης και στάθμισης των σφαλμάτων για τη διασφάλιση ακριβών και αξιόπιστων αξιολογήσεων.

## M<sup>2</sup> Scorer:

Η M<sup>2</sup> Scorer είναι ένα ειδικό εργαλείο αξιολόγησης σχεδιασμένο για το GEC. Αντιστοιχίζει τις διορθώσεις του συστήματος με τις διορθώσεις αναφοράς χρησιμοποιώντας έναν αλγόριθμο ευθυγράμμισης και υπολογίζει την ακρίβεια, την ανάκληση και το F-score. Το M<sup>2</sup> Scorer θεωρείται πρότυπο στην αξιολόγηση GEC, επειδή έχει σχεδιαστεί για να χειρίζεται τις περιπλοκές των γραμματικών διορθώσεων, συμπεριλαμβανομένων των πολλαπλών πιθανών διορθώσεων για το ίδιο σφάλμα.

## ERRANT (Error Annotation Toolkit):

Η ERRANT παρέχει ένα ολοκληρωμένο πλαίσιο για τον σχολιασμό και την αξιολόγηση των GEC. Περιλαμβάνει εργαλεία για την ανίχνευση σφαλμάτων, την ευθυγράμμιση διορθώσεων και τον υπολογισμό μετρικών, συμπεριλαμβανομένων της ακρίβειας, της ανάκλησης και του F-score. Η ERRANT προσφέρει μια πιο λεπτομερή ανάλυση της απόδοσης των GEC με την κατηγοριοποίηση των σφαλμάτων και των διορθώσεων. Αυτό επιτρέπει τη βαθύτερη κατανόηση των τύπων σφαλμάτων που το μοντέλο χειρίζεται καλά και των σημείων όπου δυσκολεύεται.

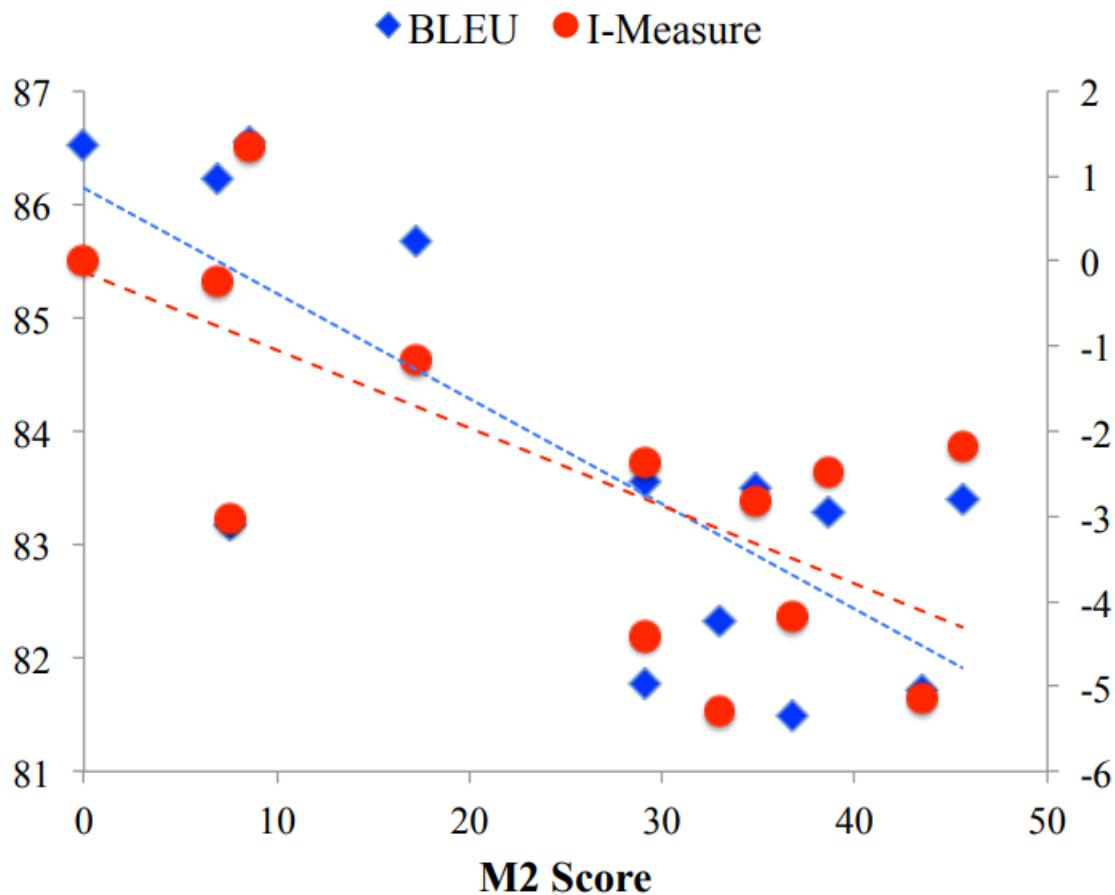
## Ανθρώπινη αξιολόγηση:

Ενώ οι αυτόματες μετρικές είναι απαραίτητες, η ανθρώπινη αξιολόγηση παραμένει ζωτικής σημασίας για την αξιολόγηση της ποιότητας των συστημάτων GEC. Οι ανθρώπινοι αξιολογητές μπορούν να παρέχουν ποιοτικές πληροφορίες σχετικά με την αναγνωσιμότητα, την ευχέρεια και τη συνολική ποιότητα του διορθωμένου κειμένου. Η ανθρώπινη αξιολόγηση συμπληρώνει τις αυτόματες μετρικές παρέχοντας μια υποκειμενική αξιολόγηση των διορθώσεων. Βοηθά στον εντοπισμό ζητημάτων που οι αυτόματες μετρικές μπορεί να παραβλέπουν, όπως οι συγκεκριμένες αποχρώσεις του πλαισίου και η συνολική συνοχή του κειμένου. Σε γενικό πλαίσιο, είναι σημαντικό να χρησιμοποιείται ένας συνδυασμός μετρικών για να υπάρχει μια ολοκληρωμένη κατανόηση της απόδοσης ενός συστήματος GEC. Ενώ η ακρίβεια, η ανάκληση και η F-score παρέχουν θεμελιώδεις πληροφορίες, μετρικές όπως η GLEU, η  $M^2$  Scorer και η ERRANT προσφέρουν πιο λεπτομερείς και διαφοροποιημένες αξιολογήσεις. Η λεπτομερής ανάλυση σφαλμάτων με τη χρήση εργαλείων όπως η ERRANT μπορεί να βοηθήσει στον εντοπισμό συγκεκριμένων περιοχών όπου το σύστημα GEC χρειάζεται βελτίωση. Για παράδειγμα, ένα μοντέλο μπορεί να έχει καλές επιδόσεις στα συντακτικά λάθη αλλά κακές στα σημασιολογικά λάθη. Επιπροσθέτως, η ενσωμάτωση της ανθρώπινης αξιολόγησης μπορεί να προσφέρει πρόσθετες γνώσεις σχετικά με τη χρηστικότητα και την απόδοχή των διορθώσεων σε πραγματικές εφαρμογές. Αυτό είναι ιδιαίτερα σημαντικό για εφαρμογές όπου η αναγνωσιμότητα και η φυσικότητα είναι κρίσιμες. Αξιοποιώντας αυτές τις μετρικές και τα εργαλεία, οι ερευνητές και οι προγραμματιστές μπορούν να αξιολογήσουν και να βελτιώσουν αποτελεσματικά τα συστήματα GEC, οδηγώντας σε πιο ακριβείς και φιλικές προς τον χρήστη λύσεις διόρθωσης γραμματικών λαθών.

Εμείς για την εργασία μας θα χρησιμοποιήσουμε μια παραλλαγή της μετρικής BLEU που ονομάζεται GLEU (Generalized Language Evaluation Understanding) και είναι ειδική για τον τύπο dataset που χρησιμοποιούμε (JFLEG).

Metrics	Calculation core	Strength	Weakness
$M^2$	Editing	<ul style="list-style-type: none"> <li>1. The calculation is relatively simple and convenient</li> <li>2. Help of CoNLL-2014 shared tasks</li> </ul>	<ul style="list-style-type: none"> <li>1. The difference between “no correction” and “error correction” cannot be captured</li> <li>2. The selective output of correction results may lead to the underestimation of system performance</li> </ul>
I-measure	Editing	<ul style="list-style-type: none"> <li>1. Multiple annotations make the systematic evaluation more objective</li> <li>2. Consider a wider range of sample extraction</li> </ul>	<ul style="list-style-type: none"> <li>1. Heavy marking workload</li> <li>2. There is no way to obtain semantic information based on the structure of the text itself</li> </ul>
GLEU	Language model	<ul style="list-style-type: none"> <li>1. No need to make detailed comments on the location and error type of each correction, and the annotation cost is small</li> <li>2. Adapt to multiple languages</li> </ul>	<ul style="list-style-type: none"> <li>1. Regardless of meaning.</li> <li>2. Without directly consider sentence structure.</li> <li>3. Cannot handle language-rich text well.</li> <li>4. Inconsistent with human judgment [25]</li> </ul>
ERRANT	Editing	<ul style="list-style-type: none"> <li>1. Low requirements for database</li> <li>2. Be able to conduct specific performance evaluation for different error types</li> </ul>	<ul style="list-style-type: none"> <li>1. Artificial rules have limited coverage of error types high postmaintenance cost</li> </ul>

Εικόνα 5.1: Σύγκριση των τεσσάρων μετρικών αξιολόγησης. Στην εικόνα παρουσιάζονται ο πυρήνας υπολογισμού, τα πλεονεκτήματα και τα μειονεκτήματα των ανωτέρω μετρικών.



Εικόνα 5.2: Συσχέτιση μεταξύ των αξιολογήσεων M2, I-measure και BLEU. Το σκορ M2 παρουσιάζει αρνητική συσχέτιση με άλλες μετρικές.

Η λανθασμένη πρόταση που γράφτηκε αρχικά αναφέρεται συνήθως ως αρχική πρόταση (source sentence) στους δείκτες αξιολόγησης των συστημάτων διόρθωσης γραμματικών λαθών. Η πρόταση που επισημάνθηκε και διορθώθηκε χειροκίνητα είναι γνωστή ως πρόταση αναφοράς (reference sentence) και η πρόταση που διορθώθηκε από το σύστημα διόρθωσης γραμματικών λαθών είναι γνωστή ως υποθετική πρόταση (hypothetical sentence). Οι χειροκίνητα επισημειωμένες και διορθωμένες προτάσεις αποτελούν το καλύτερο πρότυπο για την αξιολόγηση και χρησιμοποιούνται για τη σύγκριση με τις υποθετικές προτάσεις για την αξιολόγηση της απόδοσης του συστήματος [41].

## 5.2 Υπολογισμός Απώλειας Μοντέλου

Πριν προχωρήσουμε στην ανάλυση της αξιολόγησης του συστήματος με τη μετρική GLEU, θα αξιολογήσουμε το μοντέλο πριν και μετά τη λεπτομερή ρύθμιση χρησιμοποιώντας μια κοινή μετρική που ονομάζεται απώλεια (loss). Η απώλεια μπορεί να περιγραφεί ως το πόσο "λάθος" είναι οι προβλέψεις του μοντέλου σε σύγκριση με τις σωστές απαντήσεις. Έτσι, εάν η απώλεια μειώνεται μετά τη λεπτομερή ρύθμιση, τότε αυτό υποδηλώνει ότι το μοντέλο έμαθε. Είναι σημαντικό να χρησιμοποιούμε ξεχωριστά δεδομένα για την εκπαίδευση και την αξιολόγηση, ώστε να δείξουμε ότι το μοντέλο μπορεί να γενικεύσει την αποκτηθείσα γνώση του για την επίλυση αθέατων περιπτώσεων. Η απώλεια είναι η απλούστερη για να υλοποιηθεί με το Happy Transformer, γι' αυτό και θα τη χρησιμοποιήσουμε.

Η συνάρτηση απώλειας που χρησιμοποιείται είναι η Cross-Entropy Loss. Αυτή είναι η τυπική συνάρτηση απωλειών για μοντέλα sequence-to-sequence όπως το T5, όπου η εργασία περιλαμβάνει τη δημιουργία μιας ακολουθίας tokens (λέξεων) από μια ακολουθία εισόδου. Η Cross-Entropy Loss μετρά τη διαφορά μεταξύ δύο κατανομών πιθανότητας: της αληθινής κατανομής (η βασική τιμή αλήθειας) και της προβλεπόμενης κατανομής (η έξοδος του μοντέλου). Για εργασίες sequence-to-sequence, υπολογίζεται σε κάθε βήμα της ακολουθίας και υπολογίζεται ο μέσος όρος όλων των βημάτων.

Ακολουθεί μια πιο λεπτομερής εξήγηση του τρόπου με τον οποίο χρησιμοποιείται η Cross-Entropy Loss σε αυτό το πλαίσιο:

Απώλεια σε επίπεδο συμβόλου (Token-Level Loss):

Για κάθε λέξη στην ακολουθία-στόχο (διορθωμένη πρόταση), η Cross-Entropy Loss συγκρίνει την προβλεπόμενη κατανομή πιθανοτήτων στο λεξιλόγιο με την κωδικοποιημένη πραγματική κατανομή (την πραγματική λέξη). Αυτή η απώλεια υπολογίζεται σε κάθε χρονικό βήμα (token) στην ακολουθία εξόδου.

Απώλεια επιπέδου ακολουθίας (Sequence-Level Loss):

Οι απώλειες σε επίπεδο συμβόλου αθροίζονται ή υπολογίζονται κατά μέσο όρο σε ολόκληρη την ακολουθία εξόδου για να προκύψει η συνολική απώλεια για την εν λόγω ακολουθία.

Στόχος εκπαίδευσης:

Το μοντέλο εκπαιδεύεται για την ελαχιστοποίηση αυτής της απώλειας, δηλαδή στοχεύει στην αύξηση της πιθανότητας εμφάνισης των σωστών συμβόλων στην ακολουθία εξόδου.

Αρχικά, ας προσδιορίσουμε την απώλεια του μοντέλου στο σύνολο δεδομένων αξιολόγησης πριν από οποιαδήποτε εκπαίδευση (before training loss). Για να το πετύχουμε αυτό, θα καλέσουμε τη μέθοδο eval() του happy\_tt η οποία είναι ειδική για τον υπολογισμό απώλειας. Γενικά η μέθοδος eval() δέχεται ως εισόδους δύο ορίσματα:

- ✓ `input_filepath` (string): ένα αρχείο διαδρομής προς ένα αρχείο CSV με την ίδια μορφή που περιγράφεται για τα δεδομένα εκπαίδευσης.
- ✓ `args` (TTEvalArgs): μια κλάση δεδομένων με τα ίδια πεδία που περιγράφονται παρακάτω.

Parameter	Default
<code>preprocessing_processes</code>	1
<code>max_input_length</code>	1024
<code>max_output_length</code>	1024

Εικόνα 5.3: Παράμετροι που περιέχονται στην κλάση δεδομένων TTEvalArgs με τις default τιμές τους

Οι παράμετροι της κλάσης έχουν την ίδια σημασία με αυτές της κλάσης TTTrainArgs που περιγράφηκε παραπάνω. Η μέθοδος eval() παράγει έξοδο η οποία είναι ένα αντικείμενο με ένα μόνο πεδίο που ονομάζεται `loss`.

Για την κλήση της μεθόδου θα δώσουμε ως όρισμα τη διαδρομή προς το CSV που περιέχει τα δεδομένα αξιολόγησης (`eval.csv`), ενώ τα πεδία της κλάσης δεδομένων `args` θα παραμείνουν στις default τιμές τους.

```
before_result = happy_tt.eval("eval.csv")

Downloading data files: 100% [██████████] 1/1 [00:00<00:00, 38.19it/s]
Extracting data files: 100% [██████████] 1/1 [00:00<00:00, 46.02it/s]
Generating eval split: [██████████] 2988/0 [00:00<00:00, 46528.11 examples/s]
Map: 100% [██████████] 2988/2988 [00:00<00:00, 3323.36 examples/s]
You're using a T5TokenizerFast tokenizer. Please note that with a fast tokenizer, us:
[██████████] [2988/2988 01:38]

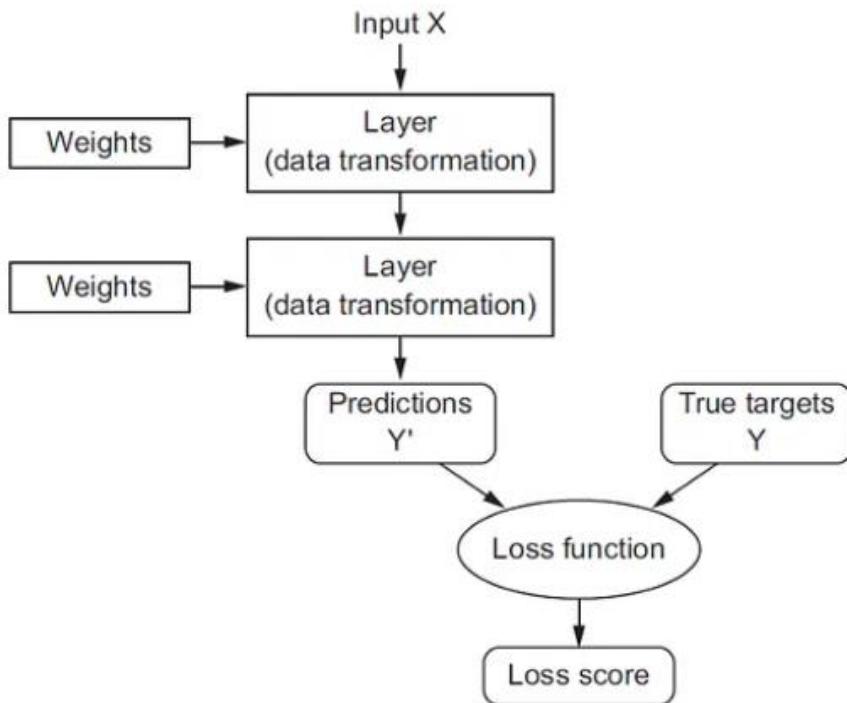
print("Before loss:", before_result.loss)

Before loss: 1.2803919315338135
```

Εικόνα 5.4: Κλήση μεθόδου για υπολογισμό απώλειας πριν την εκπαίδευση και παραγωγή αποτελέσματος

Στην εικόνα πέρα από τα αποτελέσματα της μεθόδου, διακρίνονται και τα βήματα της διαδικασίας αξιολόγησης. Πιο συγκεκριμένα, αρχικά συλλέγονται τα δεδομένα μέσα από το αρχείο που δόθηκε ως είσοδος και δημιουργείται το σώμα αξιολόγησης το οποίο είναι ολοκληρωμένο το αρχείο (eval split 2988/0). Έπειτα, χρησιμοποιείται ο T5TokenizerFast tokenizer ώστε να δημιουργηθούν τα tokens για κάθε πρόταση και να μπορέσει η μέθοδος eval() να λειτουργήσει. Εφόσον γίνουν όλα τα παραπάνω και ολοκληρωθεί η διαδικασία η μέθοδος είναι ικανή να παράγει αποτέλεσμα.

Παρατηρούμε πως η απώλεια του μοντέλου είναι αρκετά μεγάλη πριν την εκπαίδευση. Αυτό σημαίνει πως το μοντέλο δεν είναι ακόμα ικανό να προβλέψει σωστά το αναμενόμενο αποτέλεσμα, δηλαδή τις σωστές προτάσεις. Έτσι ουσιαστικά το πρόβλημα μάθησης μετατρέπεται σε πρόβλημα βελτιστοποίησης, ορίζοντας μια συνάρτηση απώλειας και στη συνέχεια βελτιστοποιώντας (εκπαιδεύοντας) τον αλγόριθμο για την ελαχιστοποίηση της συνάρτησης απώλειας [42].



Εικόνα 5.5: Διαδικασία υπολογισμού απώλειας (loss)

Έτσι λοιπόν, εκπαιδεύουμε το μοντέλο με το σύνολο εκπαίδευσης (training dataset) με σκοπό να επιτύχουμε τη βελτιστοποίηση του αλγορίθμου και τη μείωση της απώλειας. Χρησιμοποιούμε ξανά τη μέθοδο eval() για να υπολογίσουμε τη νέα απώλεια του μοντέλου.

```
after_result = happy_tt.eval("eval.csv")  
  
Map: 100% [2988/2988 00:00<00:00  
[2988/2988 01:30]  
  
print("After loss: ", after_result.loss)  
  
After loss: 0.47939983010292053
```

Εικόνα 5.6: Κλήση μεθόδου για υπολογισμό νέας απώλειας μετά την εκπαίδευση και παραγωγή αποτελέσματος

Όπως γίνεται εύκολα αντιληπτό, η απώλεια μετά την εκπαίδευση μειώνεται σημαντικά. Αυτό σημαίνει πως το μοντέλο εκπαιδεύτηκε σωστά και έχουμε επιτύχει τη βελτιστοποίηση που ψάχναμε. Συγκρίνοντας τα αποτελέσματά μας με αυτά της εργασίας του Eric Fillion που πραγματοποιεί παρόμοια υλοποίηση, παρατηρούμε πως παράγει και παρόμοια αποτελέσματα. Επομένως, επιβεβαιώνεται πως το μοντέλο μας λειτουργεί σωστά.

```
before_result = happy_tt.eval("eval.csv")
```

```
print("Before loss:", before_result.loss)
```

Result: Before loss: 1.2803919315338135

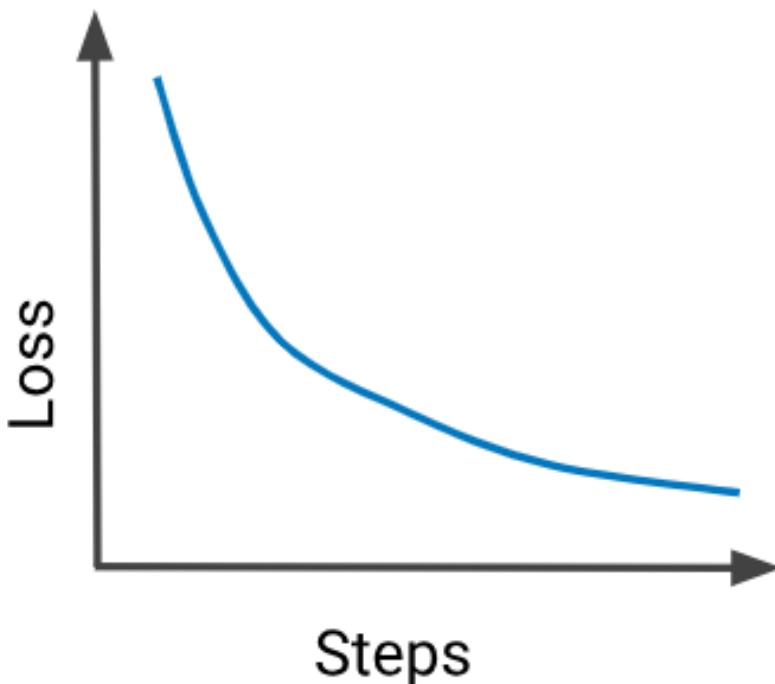
Εικόνα 5.7: Υπολογισμός απώλειας για το μοντέλο πριν την εκπαίδευση, από την εργασία του Eric Fillion

```
before_loss = happy_tt.eval("eval.csv")  
  
print("After loss: ", before_loss.loss)
```

*Result: After loss: 0.451170414686203*

*Εικόνα 5.8: Υπολογισμός νέας απώλειας για το μοντέλο μετά την εκπαίδευση, από την εργασία του Eric Fillion*

Παρατηρώντας τα αποτελέσματα της εκπαίδευσης στην εικόνα 36 αλλά και τα αποτελέσματα της αξιολόγησης υπολογίζοντας την απώλεια πριν και μετά την εκπαίδευση, διαπιστώνουμε πως σταδιακά όσο προχωράει η εκπαίδευση του μοντέλου η απώλεια συνεχώς μειώνεται μέχρι να φτάσει στην τελική της τιμή. Αυτό σημαίνει πως η απώλεια ακολουθεί έναν τύπο καμπύλης όπως παρουσιάζεται παρακάτω.



*Εικόνα 5.9: Η τιμή της απώλειας (loss) σε σχέση με τα βήματα (steps) της εκπαίδευσης.*

Από την καμπύλη φαίνεται πως η απώλεια μειώνεται ομοιόμορφα με την πάροδο των βημάτων της εκπαίδευσης, κάτι που αποδεικνύει (όπως και η μείωση της απώλειας μετά την εκπαίδευση) πως η λειτουργία του μοντέλου μας είναι σωστή. Σε περίπτωση που η καμπύλη της απώλειας δεν ήταν ομοιόμορφη και αυξομειωνόταν αρκετά, είτε η απώλεια αυξανόταν σημαντικά με την πάροδο της εκπαίδευσης, είτε η τιμή της απώλειας ήταν μεγαλύτερη μετά την εκπαίδευση, τότε το μοντέλο μας θα είχε εκπαιδευτεί λάθος και η λειτουργία του δεν θα ήταν η επιθυμητή [43].

### 5.3 Υπολογισμός GLEU score

Αφού αναλύσαμε την απώλεια (loss) του μοντέλου μας, τώρα θα εξετάσουμε την αξιολόγησή του με βάση τη μετρική GLEU. Η GLEU είναι μια παραλλαγή του μέτρου συστήματος αυτόματης μετάφρασης BLEU και παρουσιάστηκε από ερευνητές της IBM το 2001. Η GLEU χρησιμοποιείται σε μεγάλο βαθμό για την αξιολόγηση της απόδοσης των μοντέλων μηχανικής μετάφρασης (machine translation), η οποία συνήθως κυμαίνεται μεταξύ 0.0 και 1.0. Εάν οι δύο προτάσεις ταυτίζονται απόλυτα, τότε ισχύει  $\text{BLEU} = 1.0$ . Αντίθετα, αν οι δύο προτάσεις δεν ταυτίζονται σε κανένα σημείο, τότε ισχύει  $\text{BLEU} = 0.0$ . Ο πυρήνας της μετρικής αξιολόγησης της μετάφρασης BLEU είναι η ανίχνευση του αριθμού των λέξεων κοινής εμφάνισης μεταξύ των υποθετικών προτάσεων και των προτάσεων αναφοράς.

Θυμίζουμε πως η πρόταση που επισημάνθηκε και διορθώθηκε χειροκίνητα είναι γνωστή ως πρόταση αναφοράς (reference sentence) και η πρόταση που διορθώθηκε από το σύστημα διόρθωσης γραμματικών λαθών είναι γνωστή ως υποθετική πρόταση (hypothetical sentence). Η συγκεκριμένη μέθοδος υλοποίησης στηρίζεται στον υπολογισμό των n-gram της υποθετικής πρότασης και της πρότασης αναφοράς και στη συνέχεια την καταμέτρηση του αριθμού των αντιστοιχιών για την εξαγωγή της βαθμολογίας. Όσο περισσότερα grams ταυτίζονται μεταξύ της μετάφρασης του συστήματος και της χειροκίνητης μετάφρασης, τόσο υψηλότερη είναι η βαθμολογία BLEU.

Ας εξετάσουμε κάποια παραδείγματα για την κατανόηση της διαδικασίας αξιολόγησης. Τα παραδείγματα έχουν ως εξής:

- ✓ Reference Sentence: This is a small test.
- ✓ Hypothetical Sentence: This is a test.

Η παρακάτω εικόνα δείχνει ότι η βαθμολογία BLEU σύμφωνα με το 1-gram είναι 0.8, καθώς η υποθετική πρόταση μοιράζεται 4 λέξεις με την πρόταση αναφοράς. Ο αριθμός των κοινών λέξεων μεταξύ των προτάσεων (Same gram) διαιρούμενος με τις λέξεις στην πρόταση αναφοράς (Gram of reference) είναι η τελική βαθμολογία.

Κοινές λέξεις μεταξύ των προτάσεων  
Reference – Hypothetical

This – This

is – is

a – a

test – test

Διαφορετικές λέξεις μεταξύ των προτάσεων  
Reference – Hypothetical

small –

Reference	Hypothetical
-----------	--------------

This

This

Is

Is

A

A

Small

Test

Test

Same gram

4

Gram of reference

5

BLUE

0.8

Εικόνα 5.10: Τμηματοποίηση λέξεων στην πρόταση αναφοράς και την υποθετική πρόταση στο πλαίσιο του 1-gram

Αντίστοιχη διαδικασία ακολουθείται και για το 2-gram. Κάθε δύο λέξεις στην πρόταση χωρίζονται σε μια ομάδα 2-gram. Η λογική υπολογισμού είναι η ίδια με εκείνη του 1-gram. Υπό τις συνθήκες της ίδιας πρότασης αναφοράς και της υποθετικής πρότασης, η βαθμολογία BLEU είναι 0.5.

Reference	Candidate
This	This
Is	Is
A	A
Small	
Test	Test
Same gram	2
Gram of reference	4
BLUE	0.5

Εικόνα 5.11: Τμηματοποίηση λέξεων στην πρόταση αναφοράς και την υποθετική πρόταση στο πλαίσιο του 2-gram

Κοινές ομάδες 2-gram μεταξύ των προτάσεων  
Reference – Hypothetical

This is – This is  
is a – is a

Διαφορετικές ομάδες 2-gram μεταξύ των προτάσεων  
Reference – Hypothetical

a small – a  
– test

Επειδή εξακολουθεί να υπάρχει μια κρίσιμη διάκριση μεταξύ των μεταφραστικών εργασιών και των εργασιών διόρθωσης λαθών, είναι ανακριβές να θεωρείται η μηχανική μετάφραση απλώς ως μονόγλωσση μετάφραση. Η άμεση εφαρμογή της BLEU σε εργασίες GEC θα μπορούσε να οδηγήσει σε κάτι λιγότερο από ιδανικές βαθμολογίες εξόδου. Εξαιτίας αυτού, οι ερευνητές δημιούργησαν μια απλή παραλλαγή της μετρικής BLEU που ονομάζεται GLEU και είναι κατάλληλη για τις απαιτήσεις της εργασίας διόρθωσης λαθών [44].

Η ακρίβεια του συστήματος GEC υπολογίζεται μέσω της σύγκρισης της πρότασης αναφοράς και της πρότασης πηγής, δίνοντας μεγαλύτερη βαρύτητα στο gram με σωστή διόρθωση, επιβραβεύοντας το αποτέλεσμα της σωστής διόρθωσης και το σωστό κείμενο πηγής χωρίς διόρθωση, και τιμωρώντας το gram με λανθασμένη διόρθωση.

Ο μαθηματικός τύπος υπολογισμού της μετρικής GLEU έχει ως εξής:

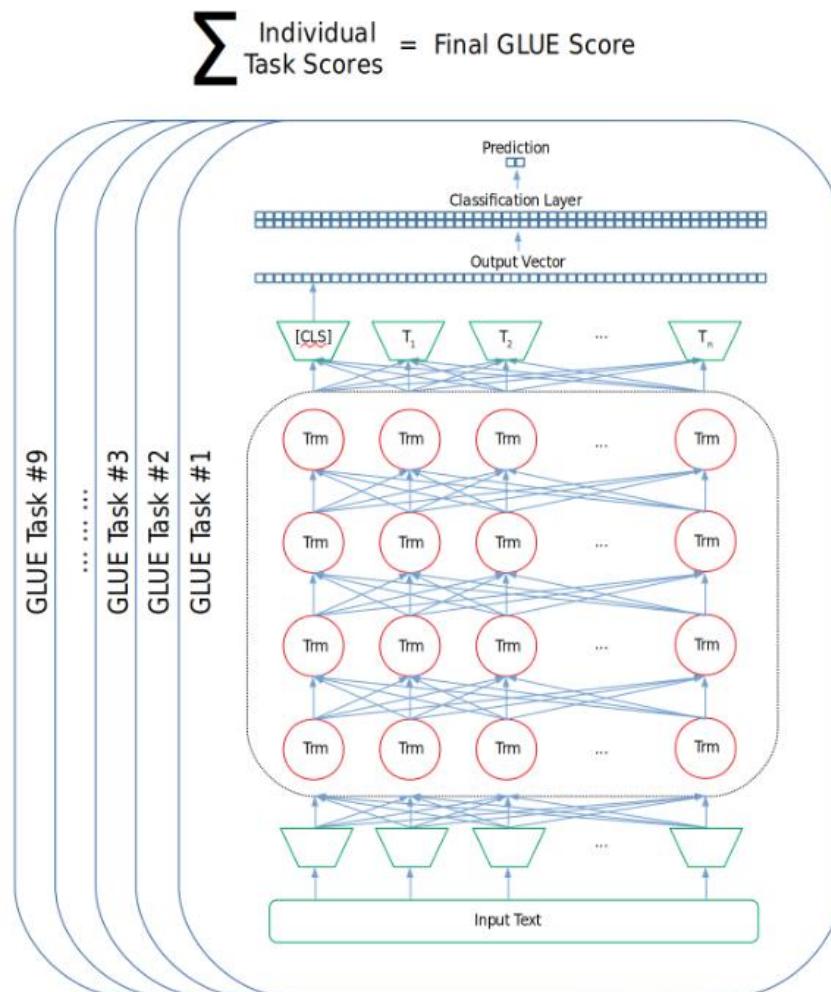
$$\text{GLEU} = \text{BP} \cdot \exp \left( \sum_{n=1}^N w_n \log p_n \right), \quad (10)$$

$$p_n = \frac{N(H, R) - (N(H, S) - N(H, S, R))}{N(H)}, \quad (11)$$

$$\text{BP} = \begin{cases} 1 & \text{if } h > r \\ \exp \left( 1 - \frac{r}{h} \right) & \text{if } h \leq r \end{cases}. \quad (12)$$

Εικόνα 5.12: Μαθηματικός τύπος υπολογισμού της μετρικής GLEU

Στους παραπάνω τύπους, ρ<sub>n</sub> είναι η ακρίβεια μετά τον υπολογισμό n-gram και BP είναι ο συντελεστής ποινής. Στον τύπο (11), H είναι το μήκος της υποθετικής πρότασης και R είναι το μήκος της πρότασης αναφοράς. Η λειτουργία του συντελεστή ποινής είναι να αποφεύγεται η μεροληψία της βαθμολόγησης του συστήματος. Κατά τη διαδικασία βαθμολόγησης, ο βαθμός ταύτισης του n-gram μπορεί να γίνει καλύτερος με τη μείωση του μήκους της πρότασης. Επομένως, για να ελεγχθεί αυτή η κατάσταση, το μήκος της πρότασης θα λαμβάνεται υπόψη στον υπολογισμό. Όταν το μήκος της υποθετικής πρότασης είναι μεγαλύτερο από αυτό της πρότασης αναφοράς ( $h > r$ ), ο συντελεστής ποινής είναι 1. Όταν το μήκος της πρότασης αναφοράς είναι μεγαλύτερο ή ίσο με το μήκος πρότασης της υπόθεσης ( $h \leq r$ ), η ποινή θα εφαρμοστεί σύμφωνα με τον τύπο. Ν είναι η τιμή του n στο n-gram του τύπου GLEU και το ανώτερο όριο του είναι 4. N(H, R) είναι τα επικαλυπτόμενα n-grams στην υποθετική πρόταση και στην πρόταση αναφοράς και N(H, S, R) είναι τα επικαλυπτόμενα n-grams στην υποθετική πρόταση (Hypothetical), στην πρόταση προέλευσης (Source) και στην πρόταση αναφοράς (Reference), αντίστοιχα. w<sub>n</sub> είναι ο σταθμισμένος μέσος όρος που υιοθετείται από το σύστημα και η τιμή του είναι 1/N.



Εικόνα 5.13: Υπολογισμός GLEU score

## 5.4 Εφαρμογή της μετρικής GLEU στο σύστημα πριν το fine-tuning

Ας δούμε λοιπόν την εφαρμογή της μετρικής στο σύστημά μας. Για να χρησιμοποιήσουμε τη μετρική πρέπει να εγκαταστήσουμε τις απαραίτητες βιβλιοθήκες. Αρχικά, πρέπει να φορτώσουμε το JFLEG test dataset από τη βιβλιοθήκη datasets χρησιμοποιώντας το import load\_dataset. Έτσι λοιπόν παίρνουμε το αρχείο test του dataset ώστε στη συνέχεια να δημιουργήσουμε προβλέψεις.

```
from datasets import load_dataset

# Load the JFLEG test dataset
dataset = load_dataset("jfleg", split='test[:]')
```

Εικόνα 5.14: Φόρτωση του συνόλου δεδομένων test για την δημιουργία προβλέψεων

Έπειτα, χρησιμοποιούμε τη βιβλιοθήκη Transformers του Hugging Face για να φορτώσουμε το μοντέλο t5-base. Ο tokenizer του μοντέλου θα μας χρησιμεύσει παρακάτω για τη δημιουργία tokens στις προτάσεις προέλευσης ώστε να παραχθούν νέες διορθωμένες (προβλέψεις).

```
from transformers import T5ForConditionalGeneration, T5Tokenizer

# Load the T5 model
model_name = "t5-base"
model = T5ForConditionalGeneration.from_pretrained(model_name)
tokenizer = T5Tokenizer.from_pretrained(model_name)
```

Εικόνα 5.15: Φόρτωση μοντέλου μαζί με τα χαρακτηριστικά του

Εφόσον έχει φορτωθεί το μοντέλο μας μπορεί να ξεκινήσει η διαδικασία για τον υπολογισμό της μετρικής GLEU. Χρησιμοποιούμε μια λίστα για τις δοκιμαστικές προτάσεις (sentences) και το μοντέλο T5 για να δημιουργήσουμε διορθωμένες προτάσεις από τις μη διορθωμένες. Η συνάρτηση generate\_predictions δέχεται ως όρισμα το μοντέλο που χρησιμοποιούμε, τον tokenizer και τις προτάσεις προέλευσης, δηλαδή τις λανθασμένες προτάσεις, και επιστρέφει κάθε φορά μια διορθωμένη. Η μεταβλητή predictions περιέχει τη λίστα με τις παραγόμενες διορθωμένες προτάσεις.

```

# Function to generate predictions
def generate_predictions(model, tokenizer, sentences):
    predictions = []
    for sentence in sentences:
        input_text = "grammar: " + sentence
        input_ids = tokenizer.encode(input_text, return_tensors="pt")
        outputs = model.generate(input_ids)
        prediction = tokenizer.decode(outputs[0], skip_special_tokens=True)
        predictions.append(prediction)
    return predictions

# Get the list of sentences from the test dataset
sentences = dataset["sentence"]

# Generate predictions
predictions = generate_predictions(model, tokenizer, sentences)

```

Εικόνα 5.16: Συνάρτηση για την παραγωγή διορθωμένων προτάσεων από τις υπάρχουσες λανθασμένες

Έχοντας πλέον στη διάθεσή μας και τις υποψήφιες προτάσεις (predictions) μπορούμε να υπολογίσουμε το GLEU score πριν το fine-tuning. Για αυτό χρησιμοποιούμε την preprocess\_references(corrections) που είναι μια συνάρτηση που λαμβάνει μια λίστα από σύνολα διορθώσεων (κάθε σύνολο περιέχει πολλαπλές διορθωμένες προτάσεις) και τα επεξεργάζεται για να χρησιμοποιηθούν για τον υπολογισμό της βαθμολογίας GLEU. Διαμέσου της συνάρτησης διασφαλίζουμε ότι περιλαμβάνονται μόνο μη κενές διορθώσεις (if correction.strip() != "") ενώ έπειτα χρησιμοποιούμε μια λίστα κατανόησης για τα tokens κάθε προβλεπόμενης πρότασης.

```

# Function to preprocess references
def preprocess_references(corrections):
    references = []
    for correction_set in corrections:
        # Each correction set is a list of corrected sentences
        formatted_references = [correction.split() for correction in correction_set if correction.strip() != ""]
        references.append(formatted_references)
    return references

# Preprocess references and predictions
references = preprocess_references(dataset["corrections"])
predictions = [pred.split() for pred in predictions]

```

Εικόνα 5.17: Δημιουργία tokens από τις προτάσεις αναφοράς και τις υποψήφιες προτάσεις

Τελευταίο βήμα αυτής της διαδικασίας είναι ο υπολογισμός της βαθμολογίας GLEU. Για αυτό το λόγο χρησιμοποιούμε την κλάση corpus\_gleu από τη βιβλιοθήκη nltk, η οποία είναι ειδική για τον υπολογισμό της μετρικής. Για τη χρήση της, χρησιμοποιούμε τα references που είναι ο κατάλογος των προτάσεων αναφοράς με χρήση συμβόλων και τα predictions που είναι ο κατάλογος των προτάσεων πρόβλεψης με χρήση συμβόλων (έξοδος του μοντέλου).

```
from nltk.translate.bleu_score import corpus_bleu

# Calculate the BLEU score
bleu_score = corpus_bleu(references, predictions)
print(f"BLEU Score: {bleu_score:.4f}")
```

Εικόνα 5.18: Υπολογισμός GLEU score για ολόκληρο το σύνολο δεδομένων δοκιμής πριν το fine-tuning

GLEU Score: 0.0945

Εικόνα 5.19: GLEU score για ολόκληρο το σύνολο δεδομένων δοκιμής πριν το fine-tuning

Όπως παρατηρούμε η τιμή της μετρικής είναι αρκετά χαμηλή, πράγμα που δείχνει πως το μοντέλο μας κατά πλειοψηφία παράγει λανθασμένα αποτελέσματα. Περίπου 9 στις 100 προτάσεις που παράγει το μοντέλο μας συμφωνούν με τις προτάσεις target του συνόλου δεδομένων αξιολόγησης. Αυτό κάνει το μοντέλο μας να μην είναι αξιόπιστο για την εργασία διόρθωσης γραμματικών λαθών. Παρακάτω θα δούμε πως μετά το fine-tuning, αυξάνεται σημαντικά η βαθμολογία GLEU του μοντέλου μας κάνοντάς το αρκετά πιο αξιόπιστο.

## 5.5 Εφαρμογή της μετρικής GLEU στο σύστημα μετά το fine-tuning

Όπως και στη διαδικασία υπολογισμού του GLEU score πριν το fine-tuning, αντίστοιχα και εδώ χρησιμοποιούμε τη συνάρτηση generate\_predictions με τη διαφορά πως πλέον δεν δέχεται ως όρισμα τον tokenizer. Επιπλέον, η λίστα για τις δοκιμαστικές προτάσεις (sentences) αποτελείται από τις προτάσεις από το eval\_dataset ενώ οι προβλέψεις γίνονται σε συνεργασία με το happy\_tt που είναι το instance του μοντέλου που έχουμε αρχικοποιήσει για μετασχηματισμούς text-to-text, συγκεκριμένα ένα μοντέλο T5 σε αυτή την περίπτωση.

```

from nltk.translate.bleu_score import corpus_bleu

# Function to generate predictions
def generate_predictions(model, sentences):
    predictions = []
    for sentence in sentences:
        input_text = "grammar: " + sentence
        result = model.generate_text(input_text)
        prediction = result.text
        predictions.append(prediction)
    return predictions

# Get the list of sentences from the test dataset
sentences = eval_dataset["sentence"]

# Generate predictions
predictions = generate_predictions(happy_tt, sentences)

# Function to preprocess references
def preprocess_references(corrections):
    references = []
    for correction_set in corrections:
        formatted_references = [correction.split() for correction in correction_set if correction.strip() != ""]
        references.append(formatted_references)
    return references

# Preprocess references and predictions
references = preprocess_references(eval_dataset["corrections"])
predictions = [pred.split() for pred in predictions]

# Calculate the BLEU score
bleu_score = corpus_bleu(references, predictions)
print(f"BLEU Score: {bleu_score:.4f}")

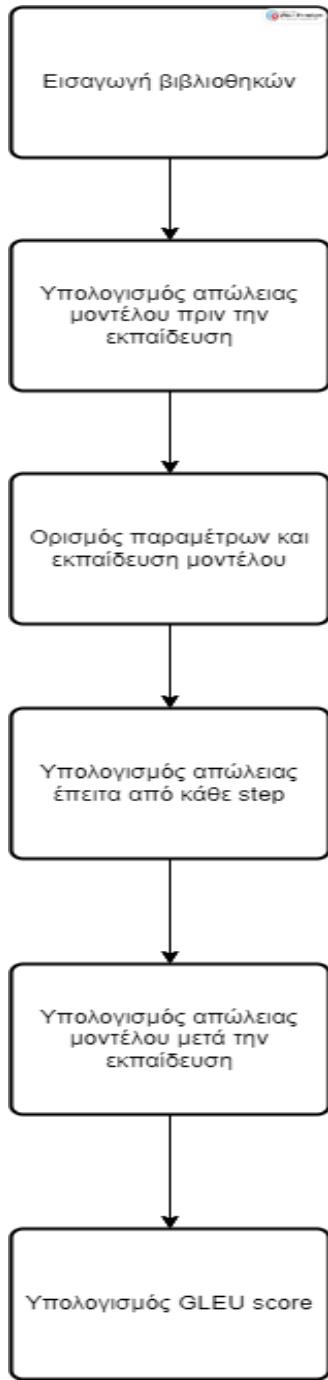
```

Εικόνα 5.20: Υπολογισμός GLEU score για ολόκληρο το σύνολο δεδομένων δοκιμής μετά το fine-tuning

**GLEU Score: 0.7553**

Εικόνα 5.21: GLEU score για ολόκληρο το σύνολο δεδομένων δοκιμής μετά το fine-tuning

Όπως παρατηρούμε η τιμή της μετρικής είναι αρκετά υψηλή, πράγμα που δείχνει πως το μοντέλο μας κατά πλειοψηφία παράγει σωστά αποτελέσματα. Περίπου 75 στις 100 προτάσεις που παράγει το μοντέλο μας συμφωνούν με τις προτάσεις target του συνόλου δεδομένων αξιολόγησης. Αυτό κάνει το μοντέλο μας να είναι αξιόπιστο για την εργασία διόρθωσης γραμματικών λαθών και να συγκαταλέγεται μεταξύ των καλύτερων πρακτικών. Παρακάτω παρουσιάζεται ένα διάγραμμα ροής για τη μεθοδολογία αξιολόγησης του μοντέλου, με όλα τα βήματα που ακολουθήθηκαν. Όλα τα τμήματα κώδικα μπορούν να βρεθούν στο [Github](#).



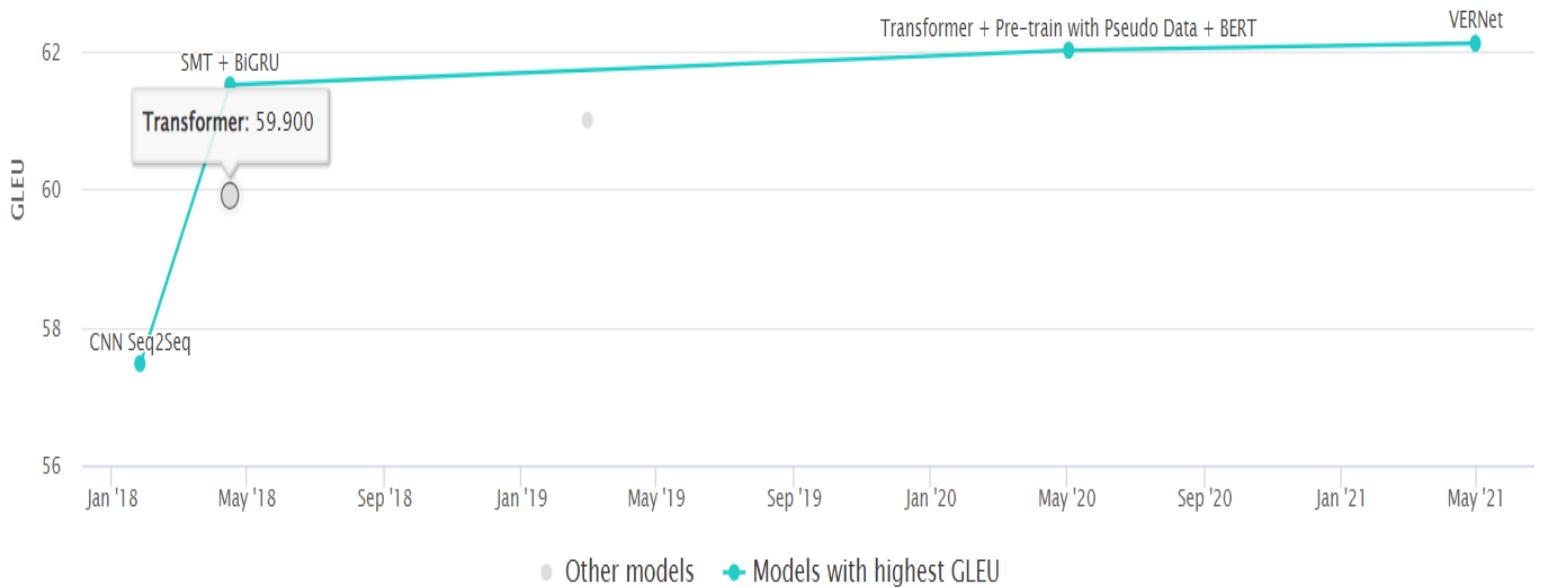
Εικόνα 5.22: Διάγραμμα ροής αξιολόγησης μοντέλου

## 5.6 Σύγκριση Αποτελεσμάτων στο GLEU benchmark

Γενικά το μοντέλο T5 παράγει υψηλές αποδόσεις στο GLEU benchmark, καταλαμβάνοντας την δέκατη θέση μεταξύ των υπόλοιπων εργασιών με σκορ 90.3 [45].

Rank	Name	Model	URL Score
1	Microsoft Alexander v-team	Turing ULR v6	 91.3
2	JDExplore d-team	Vega v1	91.3
3	Microsoft Alexander v-team	Turing NLR v5	 91.2
4	DIRL Team	DeBERTa + CLEVER	91.1
5	ERNIE Team - Baidu	ERNIE	 91.1
6	AliceMind & DIRL	StructBERT + CLEVER	 91.0
7	DeBERTa Team - Microsoft	DeBERTa / TuringNLRv4	 90.8
8	HFL iFLYTEK	MacALBERT + DKM	90.7
9	PING-AN Omni-Sinitic	ALBERT + DAAF + NAS	90.6
10	T5 Team - Google	T5	 90.3
11	Microsoft D365 AI & MSR AI & GATECH	MT-DNN-SMART	 89.9
12	Huawei Noah's Ark Lab	NEZHA-Large	89.8

Εικόνα 5.23: Πίνακας κατάταξης GLEU score εργασιών



Εικόνα 5.24: GLEU score για JFLEG dataset με χρήση διαφορετικών μεθόδων

Στην παραπάνω εικόνα βλέπουμε τα GLEU scores διαφορετικών μοντέλων που χρησιμοποιήθηκαν για την εργασία της διόρθωσης γραμματικών λαθών πάνω σε σύνολα δεδομένων τύπου JFLEG [46]. Παρατηρούμε πως χρησιμοποιώντας ένα μοντέλο που βασίζεται σε Transformer όπως το δικό μας, πετυχαίνεται απόδοση περίπου στο 60% και πιο συγκεκριμένα 59.900. Αυτό πρακτικά σημαίνει ότι το μοντέλο μας παρουσιάζει μεγαλύτερη απόδοση από άλλα μοντέλα ίδιου τύπου που χρησιμοποιούν Transformers. Και αυτό γιατί και στη δική μας εργασία πετυχαίνουμε απόδοση κοντά στο 75%, δηλαδή έως και 15% παραπάνω από τα υπόλοιπα αντίστοιχα μοντέλα.

Επιπλέον, διακρίνουμε πως η χρήση συνδυασμού μοντέλων αυξάνει το GLEU score κατά περίπου 2%. Οι συνδυασμοί μοντέλων που εμπεριέχουν τα δικά τους μοναδικά χαρακτηριστικά πιθανόν να καλύπτουν ή και να διορθώνουν ατέλειες που ενδεχομένως παρουσιάζουν τα μοντέλα μεμονωμένα. Η διόρθωση γραμματικών λαθών με τη χρήση μοντέλου που βασίζεται σε Transformer κατατάσσεται στην τέταρτη θέση του πίνακα κατάταξης GLEU score ενώ η διόρθωση γραμματικών λαθών με τη χρήση μοντέλου που βασίζεται σε Transformer + Pre-train with Pseudo Data + BERT, κατατάσσεται στην δεύτερη θέση του πίνακα κατάταξης GLEU score πάνω σε JFLEG dataset.

Rank	Model	GLEU↑
1	<b>VERNet</b>	62.1
2	<b>Transformer + Pre-train with Pseudo Data + BERT</b>	62.0
3	<b>SMT + BiGRU</b>	61.5
4	<b>Copy-augmented Model (4 Ensemble +Denoising Autoencoder)</b>	61.0
5	<b>Transformer</b>	59.9
6	<b>CNN Seq2Seq</b>	57.47

Εικόνα 5.25: Πίνακας κατάταξης GLEU score μοντέλων εφαρμοσμένων σε JFLEG dataset

Rank	Model	GLEU Score
1	Our task of GEC with Transformer	75.5
2	VERNet	62.1
3	Transformer (EncDec architecture) + Pre-train with Pseudo Data + BERT	62.0
4	SMT + BiGRU	61.5
5	Copy-augmented Model (4 Ensemble + Denoising Autoencoder)	61.0
6	Transformer (self-attention-based model by Vaswani)	59.9
7	CNN Seq2Seq	57.47

Εικόνα 5.26: Σύγκριση GLEU score της εργασίας με αντίστοιχα score μοντέλων εφαρμοσμένων σε JFLEG dataset

## 6. Συμπεράσματα

Σε αυτή τη διπλωματική παρουσιάζουμε μια έρευνα για τη διόρθωση γραμματικών λαθών (GEC) μαζί με μια ολοκληρωμένη αναδρομή στην υπάρχουσα πρόοδο. Πρώτα, δίνουμε τον ορισμό του έργου και την εισαγωγή των κοινόχρηστων συνόλων δεδομένων, του σχήματος σχολιασμού και δύο σημαντικών διαμοιρασμένων εργασιών (CoNLL-2014 και BEA-2019). Στη συνέχεια, εξηγείται λεπτομερώς ολόκληρη η διαδικασία παραγωγής κώδικα και η μεθοδολογία που ακολουθήθηκε για την εκπαίδευση του επιλεγμένου μοντέλου (T5) αλλά και για την παραγωγή αποτελεσμάτων και την αξιολόγησή τους. Διαμέσου της αξιολόγησης εξηγάγαμε κάποια χρήσιμα συμπεράσματα.

- ✓ Η εργασία μας παρουσιάζει ικανοποιητικά αποτελέσματα στη διόρθωση γραμματικών λαθών σε σχέση και με προηγούμενες εργασίες, με ποσοστό περίπου στο 75% στο GLEU benchmark.
- ✓ Η απώλεια του μοντέλου κατά την εκπαίδευση παρουσιάζει αρκετά σημαντική διαφορά, καθώς η τιμή της πριν την εκπαίδευση είναι 1.28 ενώ μετά το πέρας της εκπαίδευσης, η τιμή της καταλήγει στο 0.47. Αυτό σημαίνει πως το μοντέλο μας εκπαιδεύεται σωστά και παράγει σωστά αποτελέσματα.

- ✓ Το σκορ που πετυχαίνει η εργασία μας χρησιμοποιώντας την μετρική GLEU, ξεπερνά το σκορ πολλών εργασιών που έχουν γίνει τα προηγούμενα χρόνια με τη χρήση αρχιτεκτονικής Transformer και πλησιάζει τα επίπεδα που παρουσιάζουν άλλα μοντέλα (BERT, GPT, Hybrid approaches).

## 7. Μελλοντικές Εργασίες

Μέσα από τα συμπεράσματα που βγάλαμε από όλη την ανάλυση στην έρευνά μας, παρουσιάζουμε πέντε μελλοντικές κατευθύνσεις με βάση την υπάρχουσα πρόοδο στον τομέα της GEC.

- ✓ Προσαρμογή στην L1. Τα περισσότερα από τα υπάρχοντα έργα είναι μη προσανατολισμένα στην L1 (η πρώτη γλώσσα του συντάκτη), αντιμετωπίζοντας ως ισοδύναμα τα κείμενα που έχουν γραφτεί από συγγραφείς με διάφορες πρώτες γλώσσες. Αυτό προκαλεί την εμφάνιση μικρών αποδόσεων σε μετρικές αξιολόγησης ( $F_{0.5}$ ) που πλησιάζουν το 50% [47]. Ωστόσο, για την καλύτερη εφαρμογή της GEC, τα μελλοντικά συστήματα GEC θα πρέπει να παρέχουν πιο εξατομικευμένη ανατροφοδότηση στους Άγγλους μαθητές, λαμβάνοντας υπόψη την L1 τους. Παρόλο που ορισμένες εργασίες έχουν ήδη επικεντρωθεί σε διάφορες συγκεκριμένες πρώτες γλώσσες, υπάρχουν ακόμη πολλά περιθώρια προς διερεύνηση.
- ✓ Σενάριο χαμηλών πόρων (low-resources) για τη GEC. Τα σύνολα δεδομένων στη μηχανική μετάφραση έχουν δεκάδες εκατομμύρια ζεύγη προτάσεων, αλλά ο όγκος των παράλληλων δεδομένων στη GEC δεν είναι συγκρίσιμος ακόμη και στο μεγαλύτερο σώμα δεδομένων Lang-8 (1.147.451 προτάσεις). Το μέγεθος του σώματος δεδομένων σε άλλες γλώσσες πέραν της Αγγλικής είναι σημαντικά μικρότερο [48]. Το χειρότερο είναι ότι όταν εφαρμόζεται σε άλλες λιγότερο ομιλούμενες γλώσσες όπου δεν υπάρχει μεγάλος αριθμός παράλληλων δεδομένων, η απόδοση πολλών ισχυρών συστημάτων GEC που βασίζονται στη MT θα υποβαθμιστεί σοβαρά. Έτσι, η εκπαίδευση καλύτερων συστημάτων GEC σε σενάριο χαμηλών πόρων περιμένει να διερευνηθεί. Η πιθανή λύση μπορεί να είναι καλύτερες στρατηγικές προ-εκπαίδευσης και αύξησης των δεδομένων για την ενσωμάτωση μεγάλου όγκου κειμένων χωρίς λάθη και περισσότερη διερεύνηση προσεγγίσεων βασισμένων στη LM χωρίς την εξάρτηση από την επίβλεψη.
- ✓ Συνδυασμός διαφορετικών συστημάτων. Έχει αποδειχθεί ότι διαφορετικά συστήματα GEC είναι καλύτερα σε διαφορετικές επιδόσεις προτάσεων. Άλλωστε, το σύστημα που επιτυγχάνει την 1η και την 4η καλύτερη επίδοση στο σύνολο δοκιμών BEA-2019 βασίζεται σε σύνολο πολλαπλών μεμονωμένων συστημάτων GEC [49]. Αυτές οι αποδείξεις υποδηλώνουν ότι είναι πολλά υποσχόμενο να διερευνηθούν στρατηγικές συνδυασμού για την καλύτερη ενσωμάτωση της δύναμης των διαφορετικών συστημάτων GEC, τα οποία μπορεί να είναι εξειδικευμένα σε διαφορετικούς τύπους σφαλμάτων, θεματικές ενότητες, βαθμούς επάρκειας προτάσεων και L1.

- ✓ Σύνολα δεδομένων. Το πιο ευρέως χρησιμοποιούμενο σύνολο δεδομένων στη GEC είναι το NUCLE, το οποίο έχει απλή απόδοση προτάσεων, θεματική ενότητα και L1. Η έλλειψη ποικιλίας σημαίνει ότι η απόδοση των συστημάτων GEC σε άλλες συνθήκες παραμένει άγνωστη [50]. Εκτός αυτού, τα περισσότερα σύνολα δεδομένων περιέχουν μόνο περιορισμένο αριθμό αναφορών για τις προτάσεις προέλευσης. Περισσότερες αναφορές επιφέρουν αύξηση της συμφωνίας μεταξύ των σχολιαστών. Ωστόσο, η προσθήκη αναφορών απαιτεί επιπλέον εργασία. Αναμένεται να διερευνηθεί το πόσες αναφορές πρέπει να προστεθούν στα σύνολα δεδομένων λαμβάνοντας υπόψη το κόστος.
- ✓ Καλύτερη αξιολόγηση. Τα συστήματα αξιολογούνταν πάντα σε ένα μόνο σύνολο δοκιμών. Ωστόσο, τα διαφορετικά σύνολα δοκιμών οδηγούν σε ασυνεπείς επιδόσεις αξιολόγησης. Για την καλύτερη αξιολόγηση έχει προταθεί η διασταύρωση των συνόλων [51]. Εκτός αυτού, αν και η υπάρχουσα μετρική αξιολόγησης καταγράφει τη γραμματική διόρθωση και την ευχέρεια, κανείς δεν μετρά τη διατήρηση του νοήματος, η οποία είναι επίσης απαραίτητο να ληφθεί υπόψη κατά την αξιολόγηση ενός συστήματος GEC. Έτσι, πιο αξιόλογες μετρικές θα πρέπει να εξηγούν τη γραμματική, την ευχέρεια και την πιστότητα νοήματος της εξόδου του συστήματος.

## Αναφορές

- [1] «Grammarly,» [Ηλεκτρονικό]. Available: <https://www.grammarly.com/>.
- [2] Sagar Ailani, Ashwini Dalvi, Irfan Siddavatam, «Grammatical Error Correction (GEC): Research Approaches till now,» 2019.
- [3] Nora Madi, Hend S. Al-Khalifa, «Grammatical Error Checking Systems: A Review of Approaches and Emerging Directions,» 2018.
- [4] Yu Wang\*, Yuelin Wang\*, Jie Liu, Zhuo Liu, «A Comprehensive Survey of Grammar Error Correction».
- [5] ResearchGate, «Baseline results on CoNLL-2014».
- [6] Hwee Tou Ng, Siew Mei Wu, Ted Briscoe, Christian Hadiwinoto, Raymond Hendy Susanto, and Christopher Bryant, «The CoNLL-2014 Shared Task on Grammatical Error Correction,» 2014.
- [7] Christopher Bryant, Mariano Felice, Øistein E. Andersen, and Ted Briscoe, «The BEA-2019 Shared Task on Grammatical Error Correction,» 2019.
- [8] Thang Luong, Hieu Pham, Christopher D. Manning, «Effective Approaches to Attention-based Neural Machine Translation,» 2015.
- [9] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, Illia Polosukhin, «Attention Is All You Need,» 2017.
- [10] Ying Zhang, Hidetaka Kamigaito, Manabu Okumura, «Bidirectional Transformer Reranker for Grammatical Error Correction,» 2023.
- [11] SCRIBENDI.AI, «Grammatical Error Correction with Transformer Models,» 2021.
- [12] Marcin Junczys-Dowmunt, Roman Grundkiewicz, Shubha Guha, Kenneth Heafield, «Approaching Neural Grammatical Error Correction as a Low-Resource Machine Translation Task,» 2018.
- [13] Wei Zhao, Liang Wang, Kewei Shen, Ruoyu Jia, Jingming Liu, «Improving Grammatical Error Correction via Pre-Training a Copy-Augmented Architecture with Unlabeled Data,» 2019.
- [14] Shun Kiyono, Jun Suzuki, Masato Mita, Tomoya Mizumoto, Kentaro Inui, «An Empirical Study of Incorporating Pseudo Data into Grammatical Error Correction,» 2019.
- [15] Masahiro Kaneko, Masato Mita, Shun Kiyono, Jun Suzuki, Kentaro Inui, «Encoder-Decoder Models Can Benefit from Pre-trained Masked Language Models in Grammatical Error Correction,» 2020.
- [16] Kostiantyn Omelianchuk, Vitaliy Atrasevych, Artem Chernodub, Oleksandr Skurzhanskyi, «GECToR – Grammatical Error Correction: Tag, Not Rewrite,» 2020.
- [17] Sascha Rothe, Jonathan Mallinson, Eric Malmi, Sebastian Krause, Aliaksei Severyn, «A Simple Recipe for Multilingual Grammatical Error Correction,» 2021.

- [18] Ajit Rajasekharan, «T5 - a model that explores the limits of transfer learning,» 2019.
- [19] Qiurui Chen, «T5: a detailed explanation,» 2020.
- [20] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J. Liu, «Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer,» 2020.
- [21] «Google Colab,» [Ηλεκτρονικό]. Available: <https://colab.google/>.
- [22] «Happy Transformer Documentation».
- [23] «Hugging Face Transformers».
- [24] Eric Fillion, «Fine-Tune a Transformer Model for Grammar Correction,» 2021.
- [25] Hugging face, «T5-base model details».
- [26] Hugging face, «T5-small model details».
- [27] Hugging face, «T5-large model details».
- [28] Hugging face, «T5-3B model details».
- [29] Hugging face, «T5-11B model details».
- [30] ResearchGate, «Evaluation results of the five models with different data sizes».
- [31] Hugging face, «JFLEG dataset details».
- [32] Courtney Napoles, Keisuke Sakaguchi, Joel Tetreault, «JFLEG: A Fluency Corpus and Benchmark for Grammatical Error Correction,» 2017.
- [33] Baeldung, «Training and Validation Loss in Deep Learning,» 2023.
- [34] ResearchGate, «Training loss and validation loss of the neural network versus the number of epochs».
- [35] Hugging face, «T5-base-grammar-correction model details».
- [36] Hugging face, «GPT-2 model details».
- [37] Hugging face, «Write With Transformer».
- [38] Katerina Korre, John Pavlopoulos, «Enriching Grammatical Error Correction Resources for Modern Greek,» 2022.
- [39] Katerina Korre, Marita Chatzipanagiotou, John Pavlopoulos, «ELERRANT: Automatic Grammatical Error Type Classification for Greek,» 2021.
- [40] Courtney Napoles, Keisuke Sakaguchi, Matt Post, Joel Tetreault, «Ground truth for grammatical error correction metrics,» 2015.
- [41] Manli Long, Yan Wang, Yifei Peng, and Wanwu Huang, «A Review of the Research on the Evaluation Metrics for Automatic Grammatical Error Correction System,» Wuhan, 2022.

- [42] Aditya Rakhecha, «Importance of Loss Function in Machine Learning,» 2019.
- [43] Google Machine Learning Education, «Interpreting Loss Curves».
- [44] Chris McCormick, «GLUE Explained: Understanding BERT Through Benchmarks,» 2019.
- [45] «GLEU Leaderboard».
- [46] Papers With Code, «Grammatical Error Correction on JFLEG».
- [47] Maria Nadejde, Joel Tetreault, «Personalizing Grammatical Error Correction: Adaptation to Proficiency Level and L1,» 2019.
- [48] Jakub Náplava, Milan Straka, «Grammatical Error Correction in Low-Resource Scenarios».
- [49] Ruixi Lin, Hwhee Tou Ng, «System Combination for Grammatical Error Correction Based on Integer Programming,» 2021.
- [50] Xun Yuan, Derek Pham, Sam Davidson, Zhou Yu, «ErACoND: Error Annotated Conversational Dialog Dataset for Grammatical Error Correction,» 2022.
- [51] Masato Mita, Tomoya Mizumoto, Masahiro Kaneko, Ryo Nagata, Kentaro Inui, «Cross-Corpora Evaluation and Analysis of Grammatical Error Correction Models — Is Single-Corpus Evaluation Enough?,» 2019.