ΟΙΚΟΝΟΜΙΚΟ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΑΘΗΝΩΝ

ATHENS UNIVERSITY
OF ECONOMICS
AND BUSINESS

ΣΧΟΛΗ
ΕΠΙΣΤΗΜΩΝ &
ΤΕΧΝΟΛΟΓΙΑΣ
ΤΗΣ
ΠΛΗΡΟΦΟΡΙΑΣ
SCHOOL OF
INFORMATION
SCIENCES &
TECHNOLOGY

ΤΜΗΜΑ
ΠΛΗΡΟΦΟΡΙΚΗΣ
DEPARTMENT OF
INFORMATICS

DEPARTMENT OF INFORMATICS

M.Sc IN DIGITAL METHODS FOR THE HUMANITIES

Error Detection in English and Greek texts written by foreign learners

MSc Thesis

ELEFTHERIA STROUMPOULI

ATHENS, OCTOBER 2020

ATHENS UNIVERSITY OF ECONOMICS AND BUSINESS SCHOOL
OF INFORMATION SCIENCES & TECHNOLOGY

DEPARTMENT OF INFORMATICS

MASTER OF SCIENCE

IN DIGITAL METHODS FOR THE HUMANITIES

Error Detection in English and Greek texts written by foreign learners

MSc Thesis

ELEFTHERIA STROUMPOULI

Supervisor:	John Pavlopoulos
Reviewers:	John Pavlopoulos Ion Androutsopoulos Panos Louridas

ATHENS, OCTOBER 2020

Abstract

This thesis aims to build a system to tackle the task of detecting sentences with grammatical errors written by learners of English as a foreign language and grammatical, syntactic and semantic errors in corresponding Greek sentences. The goals of this task is to: 1) identify if the given sentence is correct or not, 2) construct a Greek corpus with artificial errors. For the second goal, real texts written by refugees and immigrants were studied as well as language exercises with deliberate mistakes in order to draw the most common mistakes that will be added to the new corpus. These mistakes were added following an algorithm, with a specific probability for all the errors, in order not to be applied in all circumstances without exception, so that the result looks more realistic. For the first goal, after the proper preprocessing of the data, three classifiers and a neural network were implemented. Logistic Regression, Support Vector Machine and Decision Tree classifiers achieved state-of-the-art scores on the English texts, while on the Greek sentence with error detected need further tuning. About the neural model (an LSTM RNN), achieved lower scores than the classifiers on the English texts and fairly good scores on the Greek texts.

Keywords: Natural Language Processing, error detection, binary classification, neural network - LSTM**,** texts written by foreign learners

Περίληψη

Η παρούσα Διπλωματική Εργασία στοχεύει στη δημιουργία ενός συστήματος που έχει ως σκοπό τον εντοπισμό αγγλικών προτάσεων με γραμματικά λάθη, γραμμένες από μαθητές της αγγλικής ως ξένης γλώσσας, και τον εντοπισμό γραμματικών, συντακτικών και εννοιολογικών λαθών σε αντίστοιχες ελληνικές προτάσεις. Ο στόχος αυτής της εργασίας χωρίζεται σε δύο υπο-στόχους: 1) ο προσδιορισμός μιας δοθείσας πρότασης εάν είναι σωστή ή λανθασμένη, 2) η κατασκευή ενός ελληνικού κειμένου με τεχνητά λάθη. Για το δεύτερο στόχο, μελετήθηκαν πραγματικά κείμενα γραμμένα από πρόσφυγες και μετανάστες, καθώς και γλωσσικές ασκήσεις που περιείχαν εσκεμμένα λάθη, προκειμένου να αντληθούν τα πιο συχνά λάθη που θα προστεθούν στο νέο κείμενο. Αυτά τα λάθη προστέθηκαν ακολουθώντας έναν αλγόριθμο, με μία συγκεκριμένη πιθανότητα για όλα, με στόχο να μην εφαρμοστούν σε όλες τις περιστάσεις ανεξαιρέτως, έτσι ώστε το αποτέλεσμα να φαίνεται πιο ρεαλιστικό. Για τον πρώτο στόχο, μετά την κατάλληλη προεπεξεργασία των δεδομένων, εφαρμόστηκαν τρεις ταξινομητές και ένα νευρωνικό δίκτυο. Οι ταξινομητές Logistic Regression, Support Vector Machine και Decision Tree πέτυχαν τελευταίας τεχνολογίας (state-of-the-art) αποτελέσματα στα αγγλικά κείμενα, ενώ στις ελληνικές προτάσεις, που είναι εντοπισμένες με λάθη, χρειάζονται περαιτέρω συντονισμό. Σχετικά με το νευρωνικό μοντέλο, το LSTM RNN, πέτυχε χαμηλότερες βαθμολογίες από τους ταξινομητές στα αγγλικά κείμενα και αρκετά καλές βαθμολογίες στα ελληνικά κείμενα.

Λέξεις-Κλειδιά: Επεξεργασία Φυσικής Γλώσσας, ανίχνευση σφαλμάτων, δυαδική ταξινόμηση, νευρωνικό δίκτυο - LSTM, κείμενα γραμμένα από ξένους μαθητές

To my family, who contributed to complete this Master.

Acknowledgements

First of all, I would like to thank my supervisor Ioannis Pavlopoulos for all the kind guidance as well as the great advice and ideas he has provided. Furthermore, I would like to extend my thanks to Ion Andtroutsopoulos and Panos Louridas for their contribution to the evaluation of this Thesis. Moreover, I am grateful to all the teachers of the Master for their support throughout the year. Additionally, I want to express gratitude to METAdrasi, a non-governmental organization, for their willingness to offer me some useful texts for my experiments. Finally, I want to thank my dear family and friends for always being there for me.

TABLE OF CONTENTS

Appendix

LIST OF TABLES

LIST OF FIGURES

FOREIGN LEARNERS’ NATIVE LANGUAGES OF THE CORPORA

- In the FCE corpus, the foreign learners had 16 different native languages. In the LANG-8 corpus, they had 65 different native languages, most being Asian languages, as the website is based in Japan.
(see https://www.cl.cam.ac.uk/research/nl/bea2019st/ for more information about the two datasets.)

- The test dataset of GFE corpus contained texts from four foreign learners who had various native languages.

Below, in the table 9, the seven most frequent native languages of the two English corpora, and the native languages of the GFE corpus are represented.

Native Language	**Corpus**		
	FCE	**Lang-8**	**GFE – test dataset**
Japanese	81	59156	0
Chinese	66	38044	0
French	146	1414	0
Spanish	200	3080	0
Italian	76	1072	0
Polish	76	1549	0
Russian	83	7159	0
Bulgarian	0	0	1
Albanian	0	0	1
Arabic	0	0	1
Turkish	0	0	1

Table 1. Most frequent foreign learners' native languages and their occurrences in the corpora.

1. Introduction

In the context of language processing and second language learning, much research have been carried out to demonstrate mechanisms, that operate on the detection of errors at the grammatical, syntactic and semantic level.

Grammatical Error Detection (GED) is a vibrant research area in Natural Language Processing (NLP). As Eisenstein [1] has mentioned, Natural Language Processing is the set of methods for making human language accessible to computers. In the last couple of years much effort has been concentrated on the detection of errors in texts written both by native speakers and by foreign language learners [2]. In learning foreign languages, grammatical error detection is applied in multiple ways, such as spelling and grammar checks and essay scoring.

In contrast to the plethora of research related to learning English as a foreign language,[1] studies about Greek language are still under-explored. Nevertheless, this thesis aims, in addition to exploring state-of-the-art techniques on grammatical error detection in English texts, to attempt to apply similar techniques on Greek texts for error detection in all three categories, grammatical, syntactic and semantic. All the texts are written by foreign learners.

In this study, supervised learning methods were used to solve the error detection task. These algorithms were trained using labeled data and, specifically, sentences with correct or incorrect labels. The employed steps were feature extraction, data processing and text classification. Some observations towards the gold errors are also reported and some conceivable rules are summarized, which might be useful for the research community. Last, the limitation of this work are analyzed and directions for improvement are proposed. Followingly: Section 2 briefly introduces the literature in this area. Section 3 indicates some details towards the datasets. Section 4 introduces the feature extraction, the learning methods that were used for the task, the experiments and the result analysis. Conclusion and reflections are arranged at last.

[1] See [2]

2. Literature Review

Systems usually identify and correct grammatical errors at the word level. This thesis, however, focuses on error detection systems at the sentence level, which are analyzed below.

1.1. Grammatical Error Correction

Grammatical Error Correction (GEC) is the occupation of correcting several types of errors in writing such as spelling, grammatical, punctuation, and word choice errors. In recent years, GEC has attracted a lot of attention, especially for English and Chinese texts, in order to develop NLP tools for learners of a language as a foreign language. The systems are often more effective for English, but they can be applied for other languages, such as German, Czech and Russian [3].

1.2. Grammatical Error Detection at token level

In general, most Grammatical Error Detection research is placed at the token level. This is because systems that take a word as input can also be used to correct errors or to identify the location of the error in the sentence.

An example of token-level GED is the token-based RNN model of Katz et al. [4] The authors used the dataset from the 2013 Conference on Computational Natural Language Learning (CoNLL), which comprises essays written by non-native learners of English and includes marked grammatical errors. Additionally, they created a synthetic data with artificial errors in order to compensate for the lack of data in the CoNLL Shared Task corpus. A trained LSTM, then, estimated a probability per token of being erroneous or not.

1.3. Grammatical Error Detection at sentence level

Concerning Grammatical Error Detection (GED) at the sentence level, an important research, by Tsai et al., [5] presents an application, the “LinggleWrite”, which supports interactive writing suggestions, scoring, error detection and corrective feedback. Grammatical suggestions, collocations, and bilingual examples were provided in order to guide the user in the direction of fluent writing. For the creation of this product, the most common grammar patterns were extracted from a corpus in order to provide writing suggestions and, also, the researchers developed models which were based on annotated learner corpora. Likewise, an existing linguistic search engine was used with the aim of delivering corrective suggestions for each error type. It must be said that the binary error tag schema, Incorrect and Correct, changed into a more informative DIRC tag schema, Delete, Insert, Replace, and Correct. A Bi-LSTM model with a Conditional Random Field layer (CRF) (Lafferty et al., 2001) was used for the training. Embeddings were created using BERT (Devlin et al., 2019) and Flair (Akbik et al., 2019), which capture more contextual information, and they were fed into the BiLSTM-CRF. About the evaluation of “LinggleWrite”, precision, recall and F0.5 were used for the grammar error detection task. The results for the Incorrect/Correct binary task showed that the model performed significantly better than the other GED models and achieved state-of-the-art performance.

Xiang et al. [6] built a system to diagnose the grammatical errors in sentences written by learners of Chinese as a foreign language with the assistance of a Conditional Random Field model (CRF). The aims of the system were to identify if this sentence contains error(s) (detection part) find the specific error types and their locations (identification part). The datasets that were used were HSK and TOCFL from 2016 and 2017 shared tasks on grammatical error diagnosis for learners of Chinese as a foreign language.[2] As for the detection part of the research, in order to optimize the scores, they deleted all the documents that had the problem of overlapping errors, they increased the amount of training data and they included syntactic features.

Li et al. [7] proposed a sentence labeling technique based on the Policy Gradient LSTM model. They also used HSK[3] as a dataset and the Word2vec tool to build Chinese word vectors. These word vectors were used to produce input sentence features. Also, features based on Parts of Speech (POS) features were used to boost the performance of the system and they generated a corresponding POS tag sequence for each Chinese sentence of the data set. Then, they used the LSTM to build a sequence labeling model. They applied reinforcement learning to map the labeling results to rewards so as to solve the problem of imbalanced positive and negative samples. This problem occurred due to the limitations of the Chinese language’s own characteristics and datasets.

Xiang et al. [8] introduced the HITSZ’s system for Chinese grammatical error diagnosis. The datasets that were used in these approaches were taken from the 2014 and the 2015 CGED Shared Tasks.[4] They experimented with POS Tri-grams and generated four classifiers, Naïve Bayes (NB), Decision Tree (DT), Support Vector Machines (SVM) and Maximum Entropy (ME), and three ensemble classifiers, Adaboost (AB), Random Forest (RF) and Random Feature Subspace (RFS). The ensembles generally behaved better than the single classifiers.

Zambieri et al. [9] created another grammatical error detection system in the context of the 2014 CGED Shared Task and they used the corpus provided by the task organizers.[5] Because of its small size, this corpus offered restricted training data, and it was challenging to create strong machine learning models for grammatical error detection. For this reason, they used a frequency-based approach to compare the produced corpus by second language learners (the learner corpus) to a journalistic corpus (the standard general language corpus) in order to filter out non-standard features of the training/test data that are more likely to be errors. This

[2] The authors did not share more specific information for the dataset.

[3] See footnote 2.

[4] The authors did not share more specific information for the dataset.

[5] The author did not share more specific information for the dataset.

approach was the easiest and the most efficient one, as it required only a large reference corpus. In addition, keyword lists were produced by comparing the two corpora. These keywords frequently revealed basic characteristics of the corpus. In this case, it can be assumed that a rational amount of these characteristics, from the learner corpus, will be infrequent distributions of words, which are very probable to be the errors. Extracted ungrammatical n-grams of the training and test corpora, which did not appear in the subset of the reference corpus, were handled as key expressions. This approach differs because they do not use the lexicon in the form of bag-of-words, but they use the total set of n-grams, 1 to 5, distracted from the corpus increasing the effectiveness of the method. With these n-gram lists, two classifiers were trained, a simple n-gram-based classifier and a Multinomial Naive Bayes (MNB) classifier to identify ungrammatical sentences.

Lee et al. [10] built a sentence classification system applying equally rule-based and n-gram-based statistical techniques on sentences written by people who learn Chinese as a foreign language (CFL). Sentences extracted from the HSK dynamic composition corpus[6] and they were manually corrected. As Chinese has no word boundaries, for the implementation of most Natural Language Processing (NLP) tasks, texts undergo by a word segmenter. A corpus-based learning method was used to merge unknown words to solve the problem of Out-Of-Vocabulary (OOV) tokens that often occur after the word segmentation process. After that, there was a POS-tagging method to label the segmented words with POS tags. Regarding the rule-based method, it provided 142 rules developed by linguistic experts to identify possible rule violations in input sentences. When an input sentence complied with a rule, the system stated the input as suspected of including grammatical errors. Regarding the n-gram statistical method (n= 2 and 3), it compared the n-gram scores of both correct and incorrect training sentences to define the correctness of the input sentences. A sentence was incorrect if its probability score by the model of incorrect sentences was higher than the correct probability. Then, these scores were used to build the respective correct and incorrect models based on a normal probability density function. Both models can, then, be applied to assess each test sentence by converting its n-gram score into a probability score to define the correctness or incorrectness of a sentence. Finally, if both methods, the rule-based and the n-gram-based, detected grammatical errors, the sentence was incorrect. For example, if the rule-based method identified that a preposition was following a verb, which is not correct for the Chinese language, and if the n-gram frequencies also showed that the frequency of this bigram (verb + preposition) is low, the sentence would be marked as incorrect.

Gupta [11] presented a rule-based approach for detecting grammatical errors made by non-native speakers of English. This method depended only on the disagreements in the outputs of two POS taggers. The POS taggers used were the Stanford Parser and the TreeTagger. The Stanford Parser employed unlexicalized Probabilistic Context-Free Grammar (PCFG) (Klein and Manning, 2003), whereas the TreeTagger used decision trees. The POS tag for each word in the data was compared with the tag given by the TreeTagger. If the number of the inferred tags for the provided sentence was not equal to the respective number of tags returned by the

[6] See footnote 2.

TreeTagger for the same sentence, as well as, if there was at least one token with different POS tags, the sentence was, then, considered grammatically incorrect.

The scores of the systems from the above studies are presented below. It is important to recall that in these metrics 100 is considered the best and 0 the worst.

AUTHORS	MODEL	PRECISION	RECALL
Tsai et al. (2020)	Bi-LSTM model with CRF+BERT	72.3%	60.6%
Xiang et al. (2018)	CRF model (Python-crfsuite)	87.46%	97.55%
Li et al (2018)	Policy Gradient LSTM model	66.98%	54.26%
Xiang et al. (2015)	Ensemble classifier RFS	50.5%	97.4%
	Ensemble learners Adaboost (AB)	71.8%	54.5%
Zambiery et al. (2014)	Simple n-gram-based classifier to identify correct (grammatical) sentences and (MNB) classifier	49.4%	77.5%
Lee et al. (2014)	Rule	85.7%	22.4%
	Rule AND 2-gram	92.4%	0.8%
Gupta (2014)	Tagger Disagreement (Stanford Parser + TreeTagger)	60.9%	33.2%

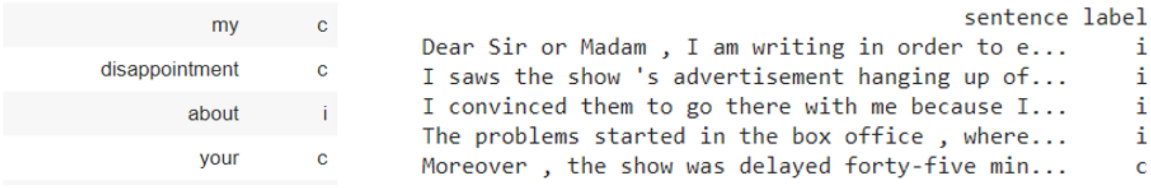
Table 2. Basic scores of the mentioned grammatical error detection systems.

3. Data Exploratory Analysis

In this chapter, the corpora used for training and testing, the process of preparing them, and their statistics will be discussed. First, the Cambridge Learner Corpus First Certificate in English (FCE) dataset and the Lang-8 Corpus of Learner English will be analyzed, drawn from the Building Educational Applications (BEA) 2019 Shared Task: Grammatical Error Correction.[7] Then, the Greek corpus will be described, which is a new corpus, created for this scope of this work. This corpus is the combination of texts written by refugees and immigrants and ready-made texts on the internet, but also a text that contains artificial errors.

3.1 The Cambridge Learner Corpus First Certificate in English

The First Certificate in English (FCE) corpus [12] is a subset of the Cambridge Learner Corpus (CLC) that contains 1,244 written answers to FCE exam questions. It was separated in train, validation and test set. In all of them, the exact errors, including their boundaries, have been manually annotated and error types have been automatically annotated with ERRor ANnotation Toolkit (ERRANT) (Bryant et al., 2017). The train set was used for training the models that will be analyzed in the next chapter and the validation set was used for the tuning of the system. The test set was used for the final evaluation. These sets were divided into two columns, one containing the words from each sentence and the other its label, "c" for correct and "i" for incorrect. For the purposes of this research, each input was reconstructed from single words into sentences as is shown in Table 2.



my	c
disappointment	c
about	i
your	c

```
                                            sentence label
Dear Sir or Madam , I am writing in order to e...     i
I saws the show 's advertisement hanging up of...     i
I convinced them to go there with me because I...     i
The problems started in the box office , where...     i
Moreover , the show was delayed forty-five min...     c
```

Table 3. On the left is the original form of the inputs and on the right the final one of FCE corpus.

Figure 1 presents the fraction of correct and incorrect sentences and it can be seen that a large proportion of the sentences has errors. From these incorrect sentences, the most used errors are stop words, as shown in Figures 2, 3 and 4. Stop words, such as “a”, “and”, “but”, are generally the most common words in a language and they are providing little to no semantic information. For this reason, they are words that are often filtered out in during the processing step of the data. In this task, however, they provide useful information for understanding errors

[7] Building Educational Applications 2019 Shared Task: Grammatical Error Correction. Retrieved October 27, 2020, from https://www.cl.cam.ac.uk/research/nl/bea2019st/

in a language, such as the fact that foreign learners have difficulties understanding how and where these seemingly easy words must be used in a sentence. Nevertheless, another distribution of errors, that does not contain stop words, was created, in Figure 2, 3 and 4 to enrich the knowledge on the errors of the texts.

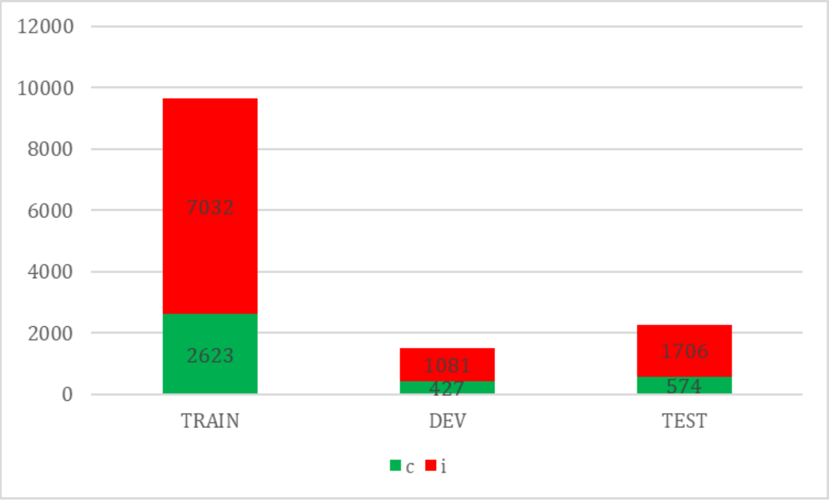


Figure 1. The distribution of correct and incorrect sentences in FCE corpus.

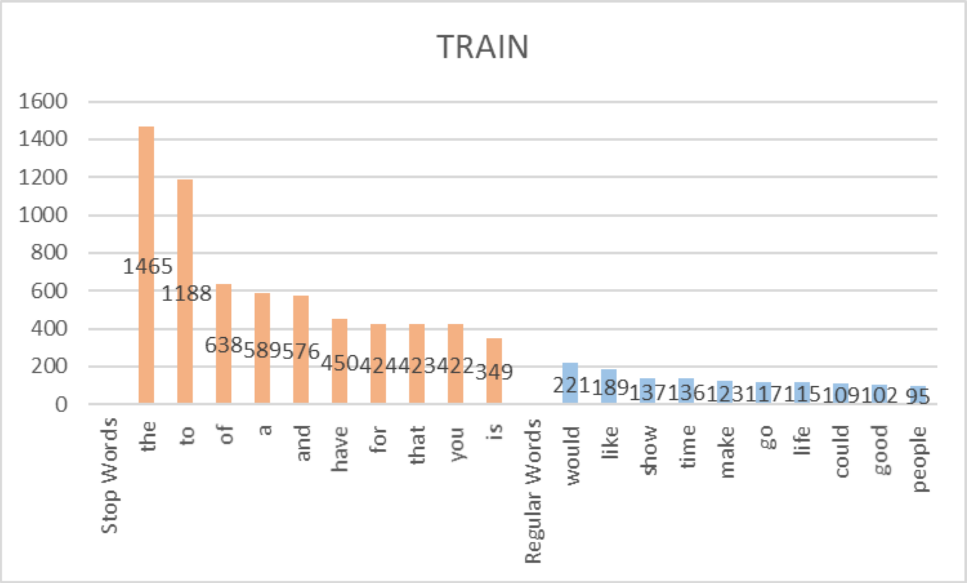


Figure 2. The distribution of stop words and regular words used most frequently incorrectly in the train set of the FCE corpus.

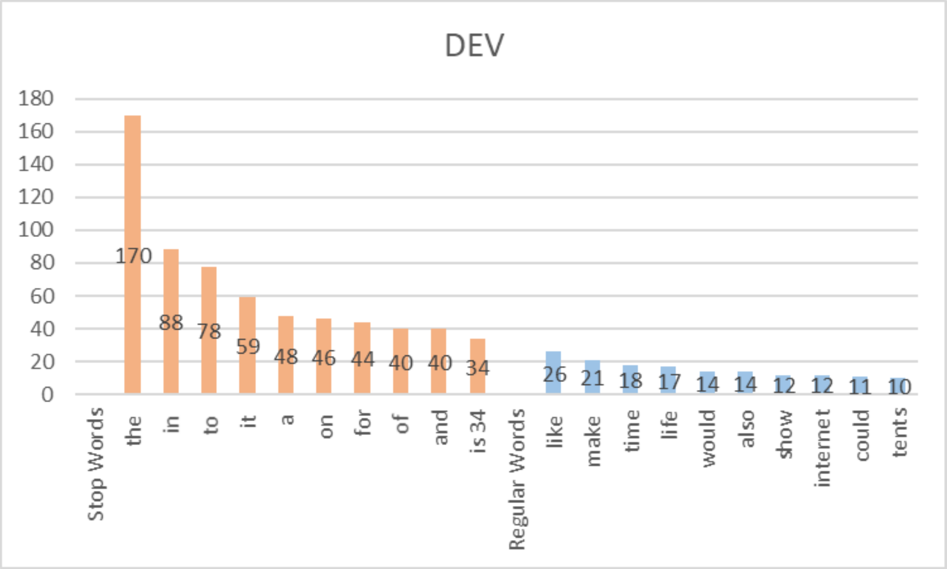


Figure 3.The distribution of the stop words and regular words used most frequently incorrectly in the dev set of the FCE corpus.

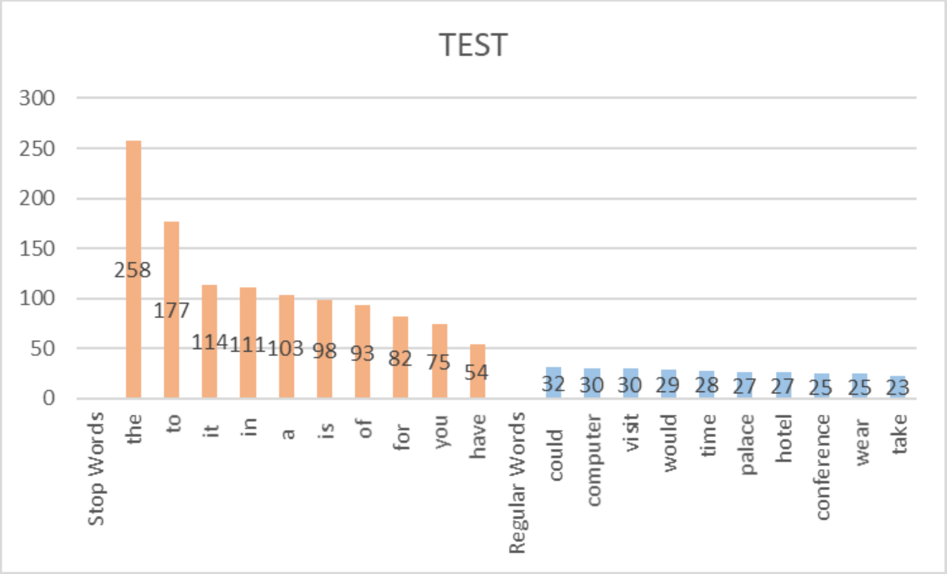
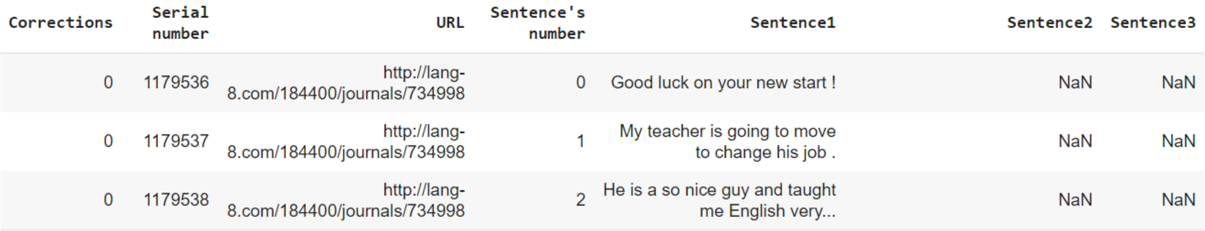


Figure 4. The distribution of the stop words and regular words used most frequently incorrectly in the test set of the FCE corpus.

3.2 The Lang-8 Corpus of Learner English

The Lang-8 Corpus [13] contains English learners texts extracted from Lang-8.[8] Lang-8 is a language learning social networking service in which language learners post their writing and native speakers correct them. The corpus has 100,051 English entries written by 29,012 active users that were divided into train and test sets. All of them had been annotated with ERRANT error types automatically. The train set was used for training the models and the test set was used for the final evaluation. These sets were separated into five or more columns, as is shown in Table 3. The columns contained a number of corrections, a serial number, the URL of the entry, the number of each sentence, the original sentence (written by a learner of English) and anything after six is the corrected sentences, if existed.



Corrections	Serial number	URL	Sentence's number	Sentence1	Sentence2	Sentence3
0	1179536	http://lang-8.com/184400/journals/734998	0	Good luck on your new start !	NaN	NaN
0	1179537	http://lang-8.com/184400/journals/734998	1	My teacher is going to move to change his job .	NaN	NaN
0	1179538	http://lang-8.com/184400/journals/734998	2	He is a so nice guy and taught me English very...	NaN	NaN

Table 4. The original form of inputs of Lang-8 corpus.

In these two datasets, the number of corrections in each sentence was counted, with the majority of the sentences comprising at most a single error (see Figure 5). These learners seem to have a better level of the language than those from the FCE dataset. Since the structure of the inputs was in a sentence level, it was very complicated to locate the incorrect words. For this reason, the words most often used in the incorrect sentences were generally counted in Figures 6 and 7, the stop words and the regular words.

[8] Multi-lingual language learning and language exchange Lang-8. Retrieved October 27, 2020, from https://lang-8.com/

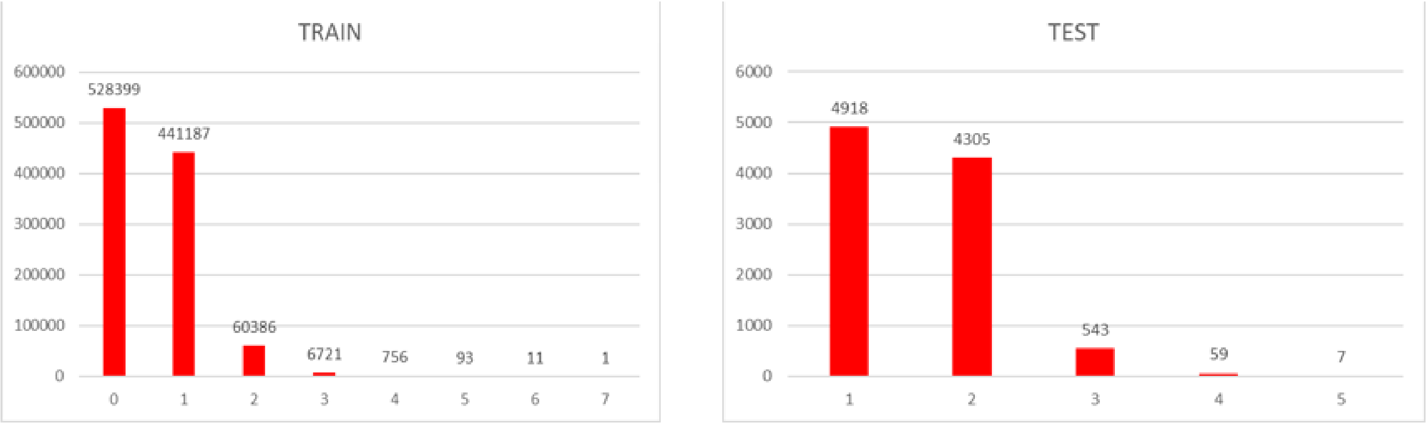


Figure 5. The distribution of the number of corrections in Lang-8 corpus.

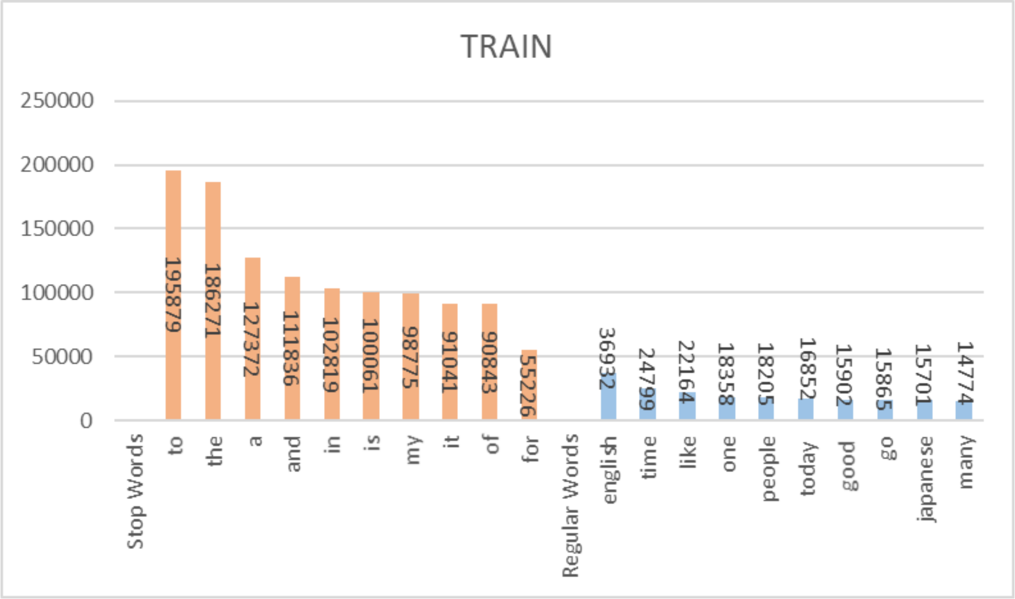


Figure 6. The distribution of the stop words and regular words used most frequently in incorrect sentences in the train set of the Lang-8 corpus.

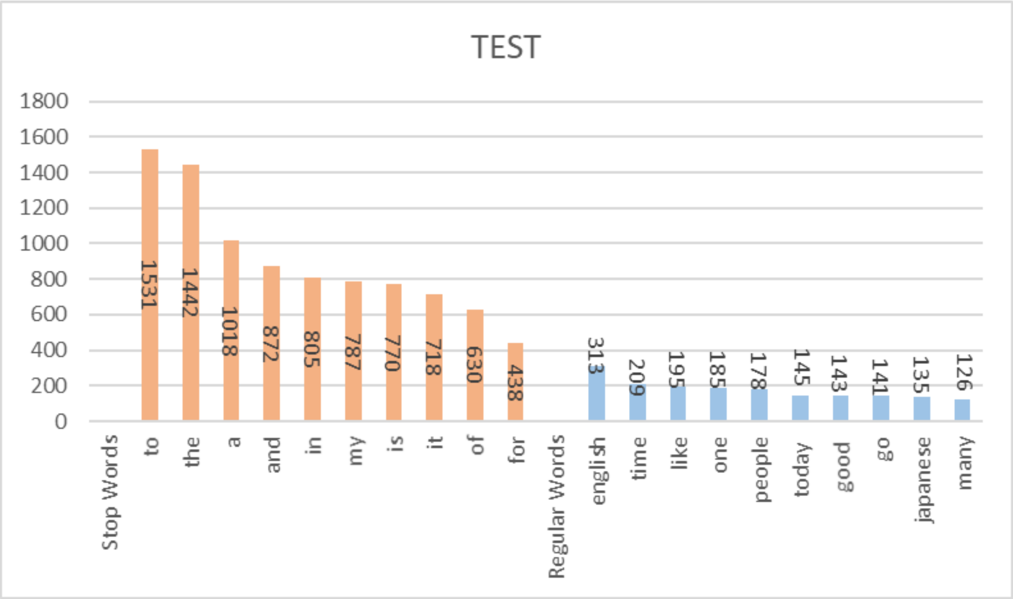


Figure 7. The distribution of the stop words and regular words used most frequently in incorrect sentences in the test set of the Lang-8 corpus.

3.3 The “Greek Frequent Errors” corpus

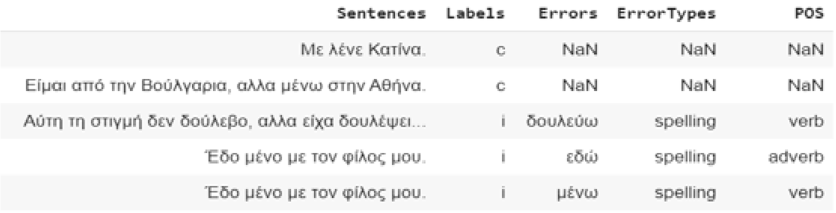
Regarding the new Greek corpus, the “Greek Frequent Errors” (GFE), it is observed that, to the best of the author's knowledge, no error detection dataset exists for the Greek language. Hence, one was built during this work. It is important to note that the reason why no such research has been performed yet for the Greek language is probably due to the complexity of the language. For example, it has various endings in noun and adjective cases. From singular to plural, it has different endings for each grammatical person and its syntax is very complicated. Table 4 displays this complexity in more details.

Cases	**Examples**
Several endings in noun and adjective cases	Nominative singular → ο άνθρωπος (human) Accusative singular → τον άνθρωπο
Various endings for each grammatical person	I read → Εγώ διαβάζω You read → Εσύ διαβάζεις
Different letters with similar pronunciation	ε,αι → [e] , η,ι,οι,ει,υ → [i] , ο,ω → [o]
Confusing syntax. Such as the verb is not always followed by the person and it can be placed anywhere in the sentence.	I called Anna → Εγώ κάλεσα την Άννα → Κάλεσα την Άννα → Την Άννα κάλεσα
Diverse intonations	ταΐζω (feed), το δωμάτιό μου (my room)

Table 5. Characteristic cases that demonstrate the complexity of Greek language.

The new corpus contains parts that were created in different ways. Initially, for the test set, in order to draw information from a realistic environment, Greek texts written by refugees, who are at a medium level of knowledge of the Greek language, were collected. These texts were offered by a non-governmental organization, “METAdrasi”,[9] which deals with refugee and immigrant issues, such as education and, specifically, learning of Greek. The texts were divided into thirty sentences and the labels correct ("c") and incorrect ("i") were annotated manually. The errors, the type per error and the POS tag of each error were annotated in the same way to construct a data frame, as it is shown in Table 5. In more detail, in the "Errors" column the correct forms of errors were filled in, in "ErrorsTypes" the types of errors were written, i.e. spelling, conjugation, syntax, and in "POS" the part-of-speech, i.e. verb, adverb, noun , preposition, article and adjective.

[9] Metadrasi – Action For Migration & Development. (2020, May 19). Retrieved October 27, 2020, from https://metadrasi.org/



Sentences	Labels	Errors	ErrorTypes	POS
Με λένε Κατίνα.	c	NaN	NaN	NaN
Είμαι από την Βούλγαρια, αλλα μένω στην Αθήνα.	c	NaN	NaN	NaN
Αύτη τη στιγμή δεν δούλεβο, αλλα είχα δουλέψει...	i	δουλεύω	spelling	verb
Έδο μένο με τον φίλος μου.	i	εδώ	spelling	adverb
Έδο μένο με τον φίλος μου.	i	μένω	spelling	verb

Table 6. The form of inputs in the test set of GFE corpus.

Then, because thirty sentences were too little for a test set, another thirty-two sentences were added from a ready-made text online. More specifically, the website "Λόγος και Επικοινωνία",[10] which focuses on the development of Greek language students' educational skills, provides exercises that help to improve students' writing. The sentences were taken from these exercises. In them, the same procedure was followed as above so that they have the exact same form. Finally, in the final test set the distribution of correct and incorrect sentences was measured (see Fig. 8), as well as, the percentage of each error type (see Fig. 9) and the distribution of part-of-speech in the incorrect sentences (see Fig. 10). It is clear that, because of the high distribution of spelling, in Figure 9, and noun and verb, in Figure 10, spelling and verb form are a very common error. That is happening because correct spelling is an ability that requires a lot of practice from the learner in every language and Greek grammar is very difficult and complicated.

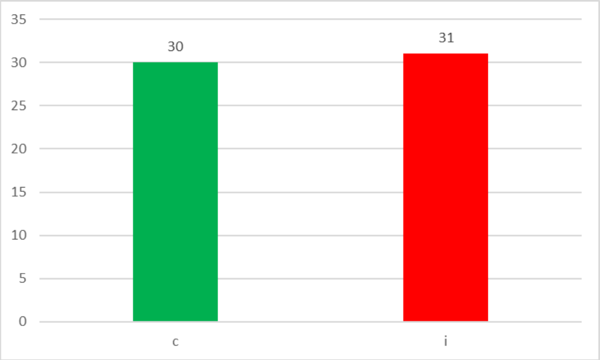


Figure 8. The distribution of correct and incorrect sentences in the test set of the GFE corpus.

[10] "Λογοθεραπεία, Εργοθεραπεία – Γλωσσικές Διαταραχές: Λόγος & Επικοινωνία." Retrieved October 27, 2020, from https://www.logosepikinonia.gr/

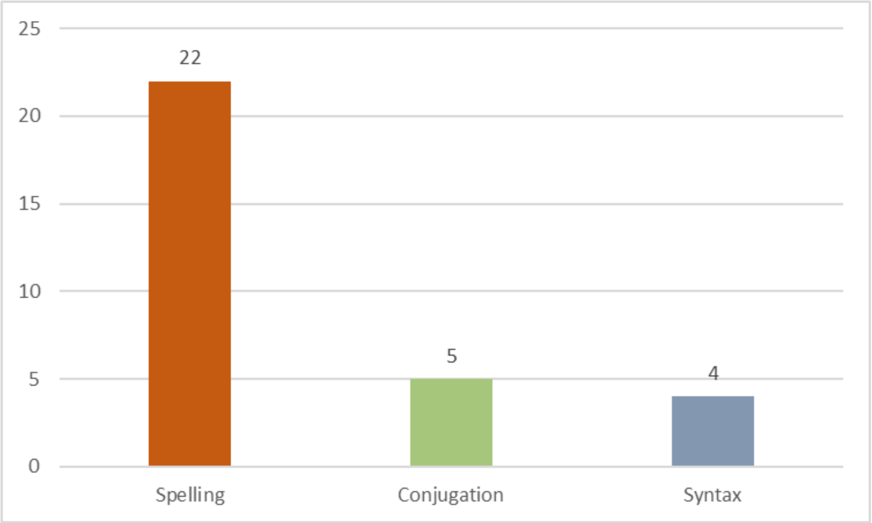


Figure 9. The distribution of error types used most frequently incorrectly in the test set of the GFE corpus.

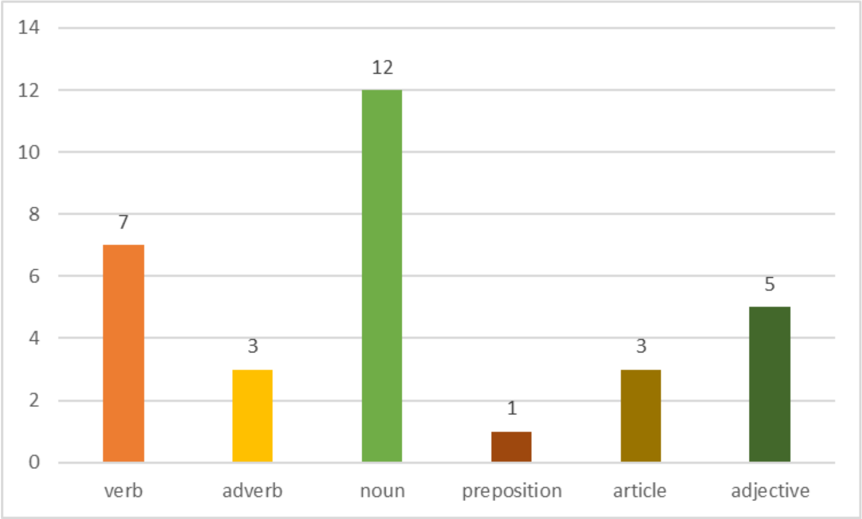
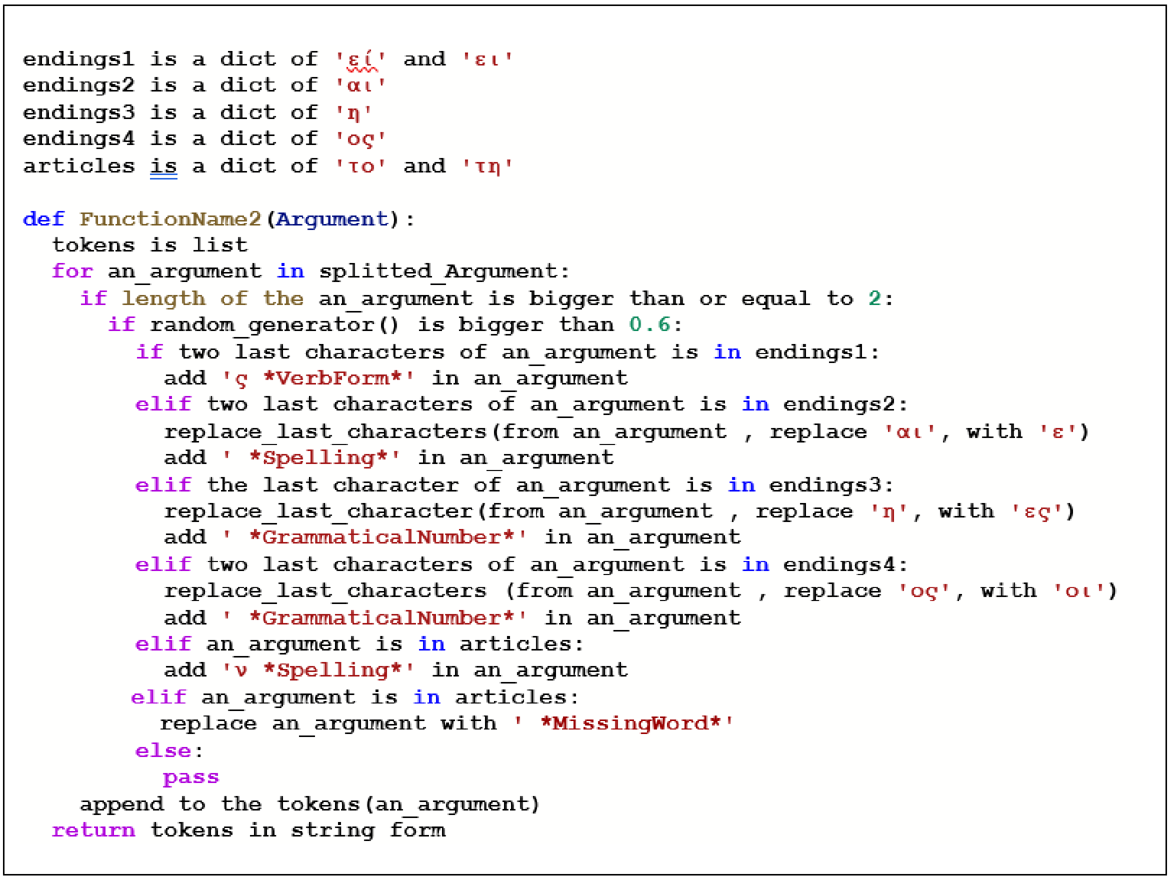


Figure 10. The distribution of the parts of speech that were used most frequently incorrectly in the test set of the GFE corpus.

For the train set, a text with artificial errors was constructed, based on the errors observed in the authentic text taken from METAdrasi. More analytically, texts of reading comprehension exercises were drawn from textbooks of learning the Greek language as a foreign language. The artificial errors were applied to these texts with the following thinking. Firstly, an attempt was made to change the grammatical form of some verbs and to replace the correct spelling with an error, such as the ending of verbs in "-αι" was changed to "-ε". These errors were labeled "VerbForm" and "Spelling" respectively. The reason behind this action was that the Greek language has different endings for each grammatical person and the spelling is very difficult for various reasons, such as the vowels with similar sound, like "η", "ι". For these reasons, a learner would have difficulty. Then, an attempt was made to change the grammatical number, such as changing the ending "-η" to "-ες" and the ending "-ος" to "οι", converting the nouns from singular to plural and vice versa. The "GrammaticalNumber" tag was added to these. These changes are justified as the Greek language has various endings in noun cases, so the learner could easily be confused. Also, some conjunctions had been removed, such as "με", meaning "with", and "σε", meaning "to", as it is observed that they are often mistakenly ignored by people who learn Greek as a foreign language. This error had been labeled "MissingWord". All the mistakes were applied randomly in the text and not in all cases, so that the result looks more realistic. The pseudocode in Table 6 shows the implementation process. Table 7 displays a summary of all the changes in the GFE corpus, namely in the “Type” column there are the four types of errors that were added, in the “Formula” there are the actual adding in the words and in the “Probability” there is the probability that was applied in all of them.



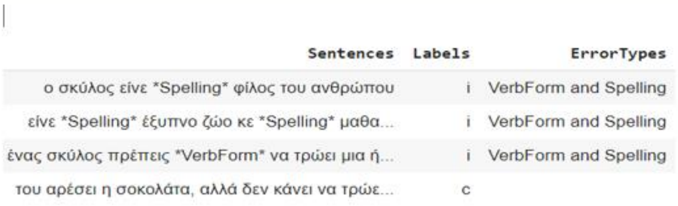
```
endings1 is a dict of 'εί' and 'ει'
endings2 is a dict of 'αι'
endings3 is a dict of 'η'
endings4 is a dict of 'ος'
articles is a dict of 'το' and 'τη'

def FunctionName2(Argument):
  tokens is list
  for an_argument in splitted_Argument:
    if length of the an_argument is bigger than or equal to 2:
      if random_generator() is bigger than 0.6:
        if two last characters of an_argument is in endings1:
          add 'ς *VerbForm*' in an_argument
        elif two last characters of an_argument is in endings2:
          replace_last_characters(from an_argument , replace 'αι', with 'ε')
          add ' *Spelling*' in an_argument
        elif the last character of an_argument is in endings3:
          replace_last_character(from an_argument , replace 'η', with 'ες')
          add ' *GrammaticalNumber*' in an_argument
        elif two last characters of an_argument is in endings4:
          replace_last_characters (from an_argument , replace 'ος', with 'οι')
          add ' *GrammaticalNumber*' in an_argument
        elif an_argument is in articles:
          add 'ν *Spelling*' in an_argument
        elif an_argument is in articles:
          replace an_argument with ' *MissingWord*'
        else:
          pass
    append to the tokens(an_argument)
  return tokens in string form
```

Table 7. Pseudocode of adding errors in our GFE corpus.

Type	**Formula**	**Probability**
Verb Form	add -ς	>0.6
Grammatical Number	-η → -ες	
	-ος → -οι	
Spelling	-αι → -ε	
	add -ν on articles	
Missing Word	remove με, σε	

Table 8. All the possible changes that were applied to create GFE.

This text was split into 301 sentences, which were mechanically annotated as follows. When a sentence contained one of the error tags, it was annotated with "i" for incorrect, otherwise with "c" for correct. Also, the error types of each sentence were added. Table 8 shows the final form of this annotations that were used as inputs.



Sentences	Labels	ErrorTypes
ο σκύλος είνε *Spelling* φίλος του ανθρώπου	i	VerbForm and Spelling
είνε *Spelling* έξυπνο ζώο κε *Spelling* μαθα...	i	VerbForm and Spelling
ένας σκύλος πρέπεις *VerbForm* να τρώει μια ή...	i	VerbForm and Spelling
του αρέσει η σοκολάτα, αλλά δεν κάνει να τρώε...	c	

Table 9. The form of inputs in the train set of the GFE corpus that includes artificial errors.

In these sentences the correct and incorrect labels (Fig. 11) and the most common error types, Figure 12, were calculated to create an overview of the data.

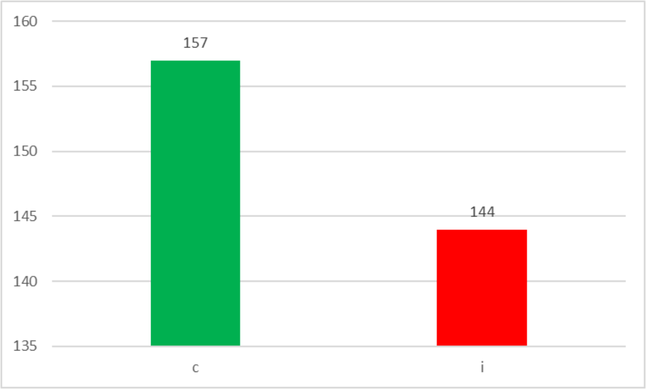


Figure 11. The distribution of correct and incorrect sentences in the train set of the GFE corpus with artificial errors.

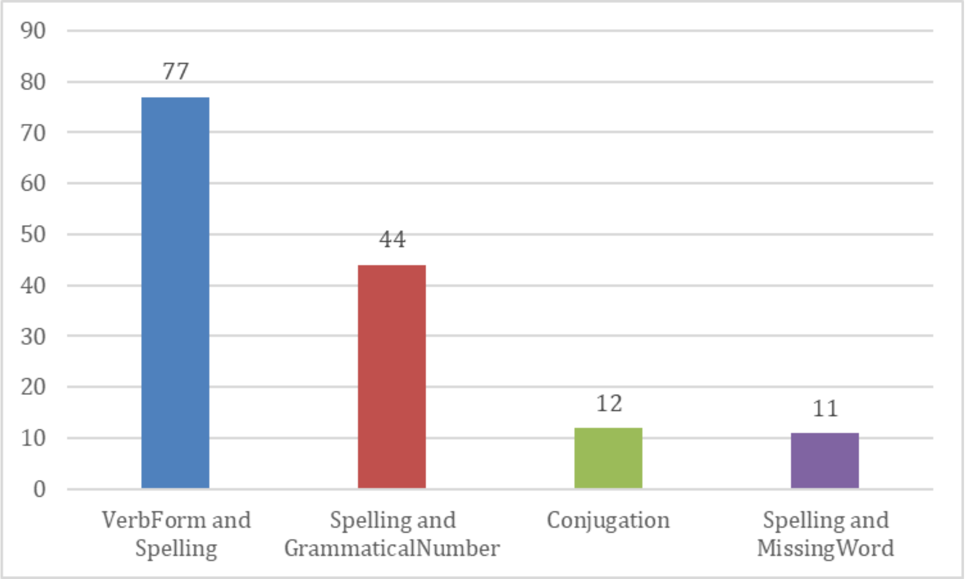


Figure 12. The distribution of error types used most frequently incorrectly in the train set of the GFE corpus with artificial errors.

4. Experiments

A series of experiments was conducted, in all three datasets, in order to explore the robustness and the possible improvements of the concerned detection systems. Three supervised learning classifiers, namely Logistic Regression, Support Vector Machines and Decision Tree, and a Long Short-Term Memory (LSTM) RNN (Subasi, 2020). The details and results of these experiments will be discussed in this chapter.

4.1. Method

Regarding to the three first classifiers, for their application, it was necessary to preprocess the data properly, vectorizing the sentences and transforming the non-numerical labels to numerical. These procedures were performed automatically with the help of TfidfVectorizer and LabelEncoder functions.[11] The input labels for the Logistic Regression and Decision Tree classifiers were "1" and "0" for incorrect and correct labels, respectively. On the other hand, in the case of Support Vector Machine, the labels were converted to "-1" and "1". At this stage, the columns of sentences and labels from the train and the test set were used. For the implementation of the classifiers, firstly, they were trained with the vectorized training data and their labels and, secondly, they made predictions with the vectorized testing data and their labels.

After that, the effort to improve the ability of these classifiers followed, by tuning their hypermeters. To achieve this, Grid search was used, which is a library function that is a member of sklearn's `model_selection` package,[12] and the columns of sentences and labels from the validation set were used. Grid search is the process of performing hyper parameter tuning in order to determine the optimal values for a given model. With its help, the best-chosen set of hyper parameter values were used in the current model. Their evaluation will be discussed in the next subsection.

For the application of the Long Short-Term Memory model (Hochreiter, 1997), it was necessary to vectorize the sentences and transform the non-numerical labels to numerical. To achieve this, the words of every sentences were converted into indices, by getting their keys from a sorted dictionary. It is important to note that in this dictionary, the first 3 integer positions are reserved, with "PAD" and "UNK". PAD was used in order to complement small texts and reach the max length and UNK was used to mask the words that were discarded from the vocabulary, such as stop words. Also, the label of the correct was converted to the number 0, while the label of the incorrect to the number 1.

The LSTM's model was then constructed, with layers of a 128-dimensional embedding, 2 LSTMs with 64 internal units each one and a sigmoid activation function as a classifier, such

[11] https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html and https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.LabelEncoder.html

[12] https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html

as in Figure 14. The sigmoid was chosen as it results in a binary value, between 0 and 1, which indicates how confident the model is of the example being in the class.

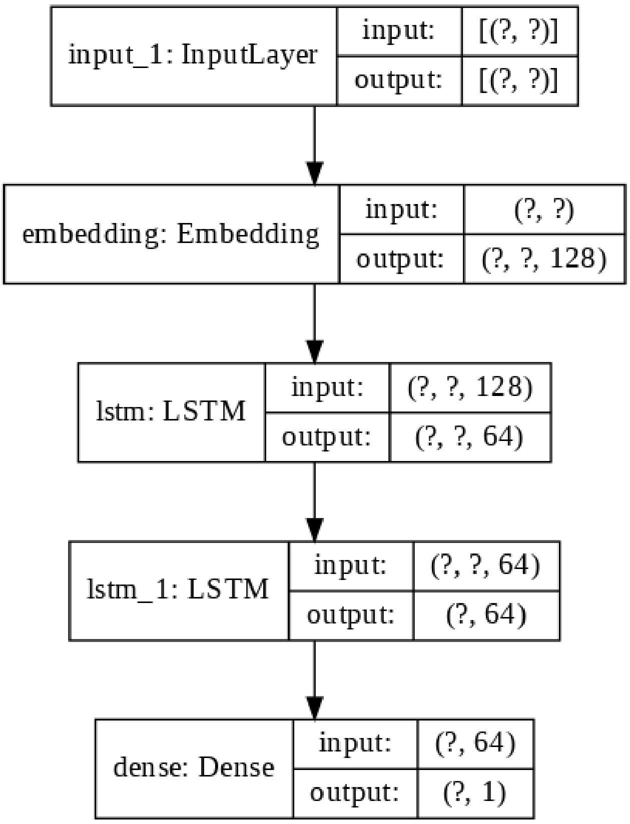


Figure 13. Architecture of the LSTM model.

Then, early stopping criterion, a form of regularization, was added in order to not wait for the best epoch to arrive. Early stopping monitors the chosen metric and stops the training of the model when no gains are monitored, above some patience threshold. The patience parameter was determined to 5 and Area under the ROC Curve (AUC) was monitored on the validation set. The model was compiled with Adam optimizer, a replacement optimization algorithm ,[13] and Binary Cross Entropy in the loss function. After preliminary experiments, the learning rate of the optimizer was determined to 0.001 for the FCE dataset and to the default value, 0.01, for the LANG-8 and the Greek datasets. Binary Cross Entropy is a loss function that is used in binary classification tasks, quantifying the difference between two probability distributions. Finally, the model made predictions, after being trained with preprocessed data and labels and some parameters were added, such as batch size to 128, the number of samples to work through before updating the internal model parameters, and epochs to 100 for the FCE corpus and 10 batch size with 100 epochs for LANG-8 corpus. For the GFE corpus batch size was 10 and epochs 50. All the parameters are mentioned in the Table 8. It is notable that three

[13] See https://keras.io/api/optimizers/adam/

different seeds were used and their average, ensemble, score was measured for the final result. The model's evaluation will be discussed in the next subsection.

Dataset	LSTM model Parameter	Range	Dataset	LSTM model Parameter	Range	Dataset	LSTM model Parameter	Range
FCE	Batch size	128	*LANG-8*	Batch size	10	*GFE*	Batch size	10
	Epochs	100		Epochs	100		Epochs	50
	Embedding dim	128		Embedding dim	128		Embedding dim	128
	Output activation function	Sigmoid		Output activation function	Sigmoid		Output activation function	Sigmoid
	Loss function	Binary Cross Entropy		Loss function	Binary Cross Entropy		Loss function	Binary Cross Entropy
	Optimizer	Adam		Optimizer	Adam		Optimizer	Adam
	Learning rate	0.001		Learning rate	0.01		Learning rate	0.01

Table 10. The chosen parameters applied to the LSTM model.

4.2. Evaluation and Analysis

The evaluation of the classifiers was done through Precision, Recall, Area Under the Receiver Operating Characteristic Curve (ROC AUC), F0.5 score and Area under Precision-Recall Curve (AUPR).

Regarding the performance of the classifiers on the FCE and LANG-8 datasets, the results are shown in Figures 16 and 17, respectively. It is observed that the final scores seem to be comparable to the state-of-the-art performance for the detection task, namely the scores of Table 1. More specifically, about the FCE dataset, the SVM, the Logistic Regression and the Decision Tree had a slightly better performance than the LSTM model. In fact, the Logistic Regression had the best scores in Precision, and F0.5. The Decision tree had the highest scores in Recall, meaning that it had great ability to find most of the correct sentences, and the SVM in AUPR. The LSTM performed better in ROC AUC than the others.

About the LANG-8 dataset, the SVM, the Logistic Regression and the Decision Tree had better scores than the LSTM model. The Logistic Regression had the greatest scores except from ROC AUC, being the most suitable model. In ROC AUC, the SVM had a little higher performance and, in general, it had the second-best results.

One possible reason that the LSTM had lower scores is overfitting. That is, the function might be too closely fit the set of data points. This will result in the performance on the train set being good and continuing to improve, whereas performance on the validation set improves to a point and then begins to degrade. A possible solution could be to add F05 as monitoring function in early stopping. Some others additional solutions might be the addition of dropout layers, weight regularization and an attention mechanism, but also the appropriate tuning of the hypermeters of early stopping and fit function.

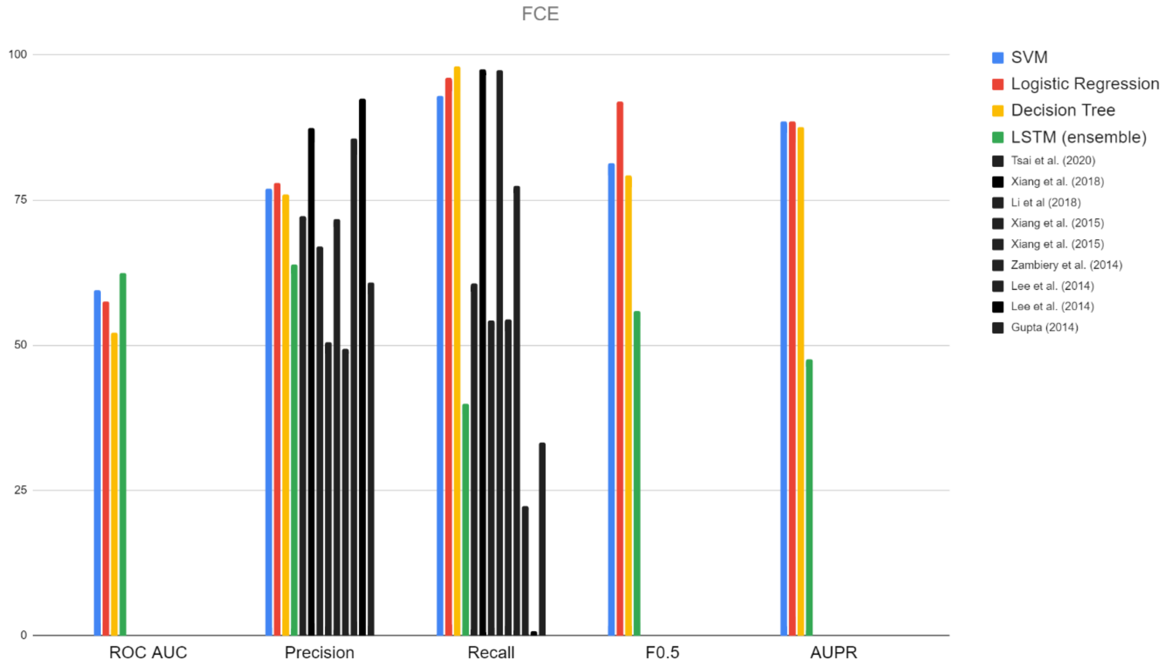


Figure 14. The final percentage results of the four classifiers on FCE data set with the state-of-the-art scores.

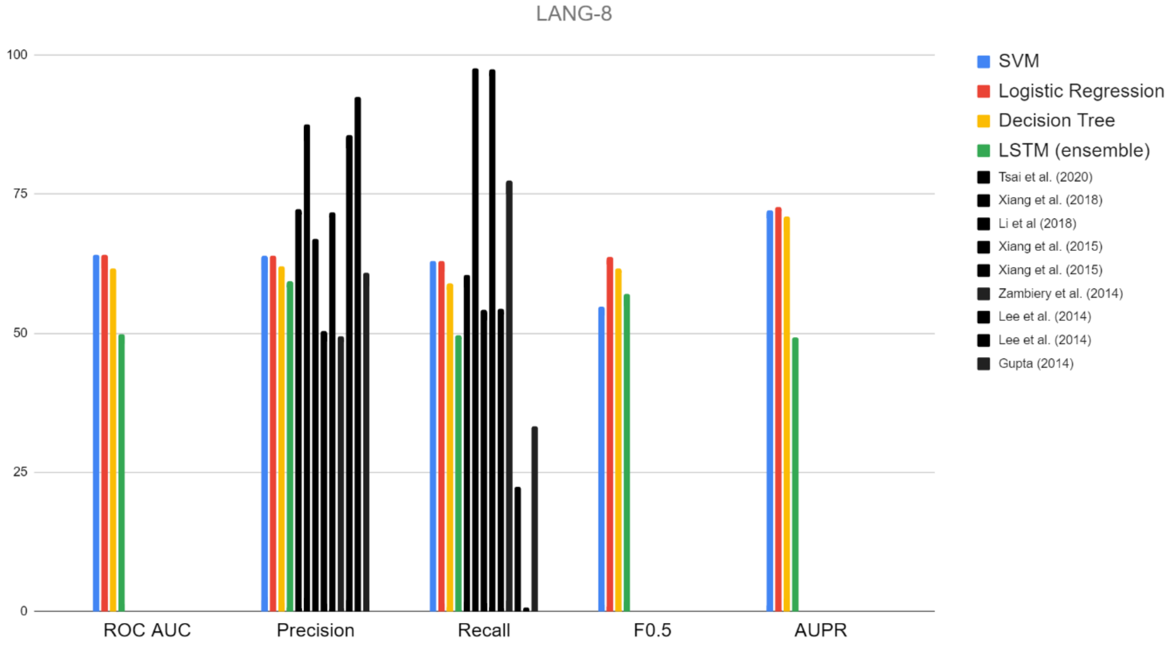


Figure 15. The final percentage results of the four classifiers on LANG-8 data set with the state-of-the-art scores.

Finally, the results on the GFE corpus are shown in Figure 18. It is important to recall that the test dataset contains real-world errors and it is only the training dataset that comprises linguistic rules. For this corpus, it is observed that the neural model had a better performance than for the other two datasets. It had the best scores in ROC AUC, Recall and F05. The Decision tree had the highest scores in Precision and AUPR. For this reason, these two models were the most qualified for our task.

As a conclusion for the Greek corpus, there is, definitely, room for improvement, such as adding more suitable or more advanced functions to search for the best parameters of the classifiers and the aforementioned tuning for the LSTM model. Nevertheless, concerning that the Greek language has not been, sufficiently, used in the area of natural language processing, and, especially, in error detection, the results are acceptable.

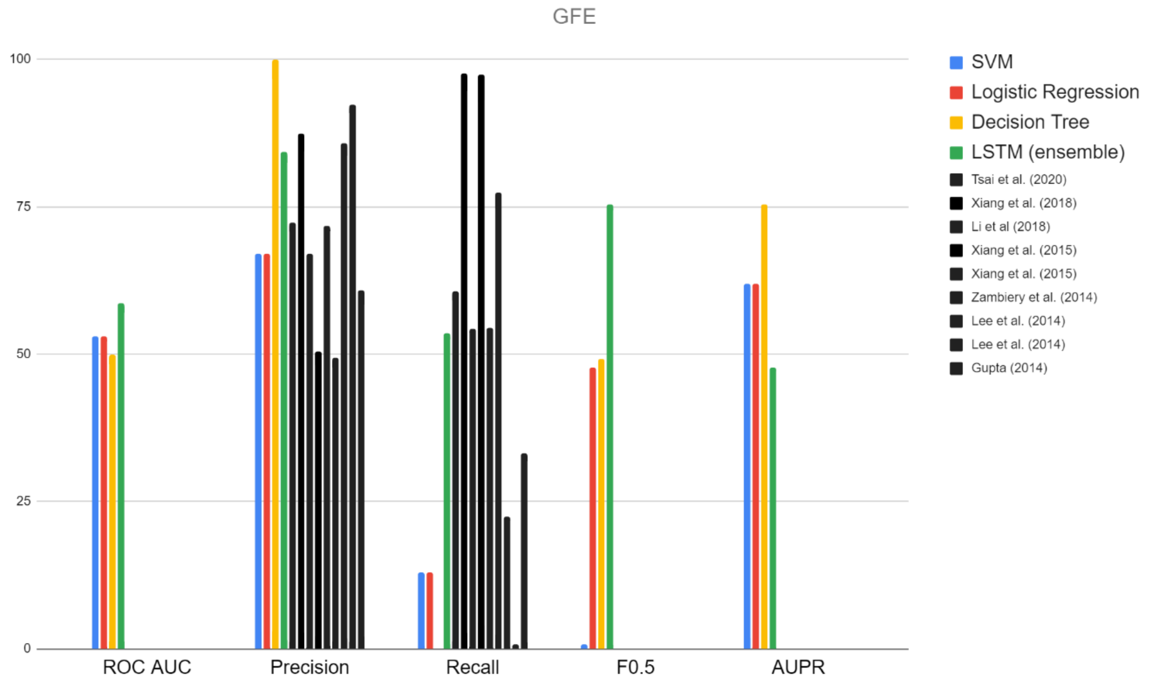


Figure 16. The final percentage results of the four classifiers on the GFE corpus with the state-of-the-art scores.

DATASET	CLASSIFIER	ROC AUC	PRECISION	RECALL	F0.5	AUPR
FCE	**SVM**	59.6	77	93	81.4	88.6
	Logistic Regression	57.5	78	96	92	88.5
	Decision Tree	52.2	76	98	79.3	87.6
	LSTM (ensemble)	62.4	64	40	56	47.6
LANG- 8	**SVM**	64.2	64	63	54.8	72.2
	Logistic Regression	64.1	64	63	63.7	72.6
	Decision Tree	61.7	62	59	61.6	71
	LSTM (ensemble)	49.8	59.4	49.7	57	49.2
GFE	**SVM**	53.1	67	13	0.7	61.9
	Logistic Regression	53.1	67	13	47.7	61.9
	Decision Tree	50	100	0	49.2	75.4
	LSTM (ensemble)	58.6	84.4	53.5	75.5	47.8

Table 11. The final percentage scores of the four classifiers on the three corpora.

5. Conclusion and Reflections

This thesis is focused on dealing with the task of detecting English sentences with grammatical errors and Greek sentences with grammatical, syntactic and semantic errors. All the texts are written by foreign learners or they have been constructed based on their writing. The task can be divided into two subtasks. The first, main, subtask is binary classification, or in other words, to identify whether a sentence contains error(s) or not, and the second subtask is to construct a new Greek corpus with artificial errors, as it is a novel project. The first one is relied on similar tasks about error detection, while the second is an original technique, based on linguistic rules.

Regarding the construction of the Greek corpus, texts written by refugees and immigrants and ready-made texts with errors found on the internet were combined and studied in order to create a new bigger corpus. By the observations of the original errors, artificial errors were added in other texts drawn from textbooks of learning the Greek language as a foreign language. The artificial dataset was meant to be used for the system training and the text with the real-world errors was used as the test dataset for the evaluation. The final form of each dataset was annotated with the errors of each sentence.

Regarding the detection task, after preprocessing the data, a method with four different classifiers was followed. The three classifiers were Logistic Regression, Support Vector Machine and Decision Tree and the other was an LSTM model. The Logistic Regression was the one that had the best scores for the English corpora, and the Decision Tree for the Greek corpus. The LSTM model had worse scores than the other classifiers for the two English texts, because of the possible overfitting, but for the Greek texts had greater performance.

The above results of the detection system, apart from its weak spots, achieved state-of-the-art performance for the English language and, as for the Greek language, the scores were fairly good concerning the novel task. Notwithstanding this fact, there is scope for improving, such as the addition of dropout layers, weight regularization and an attention mechanism or the appropriate tuning of all the hypermeters.

Bibliography

[1] Eisenstein, J. (2019). Introduction to Natural Language Processing. Amsterdam University Press., p.1

[2] K. M. Knill, M. J. F. Gales, P. P. Manakul and A. P. Caines. (2008). Automatic Grammatical Error Detection of Non-native Spoken Learner English, In International Conference on Acoustics, Speech and Signal Processing (ICASSP), Brighton, United Kingdom, pp. 8127-8131; Felice, R.D. (2008). Automatic error detection in non-native English [Master's thesis, University of Oxford].

[3] Katsumata, S., & Komachi, M. (2020). Stronger Baselines for Grammatical Error Correction Using Pretrained Encoder-Decoder Model.

[4] Katz, B., Plant, R., Kuklisova, N., (2016). Essay Scoring with Grammatical Error Detection. UC Berkeley School of Information, pp. 1-10

[5] Tsai, Ch., Chen, J., Yang, Ch., Chang, J. (2020). LinggleWrite: a Coaching System for Essay Writing. Association for Computational Linguistics, p. 127-133

[6] Xiang, Y. (2018) Grammatical Error Identification for Learners of Chinese as a Foreign Language [Master's thesis, Uppsala University].

[7] Li, Ch., Qi, J.(2018). Chinese Grammatical Error Diagnosis Based on Policy Gradient LSTM Model. Association for Computational Linguistics, pp. 77-82

[8] Xiang, Y., Wang, X., Han, W., Hong, Q. (2015). Chinese Grammatical Error Diagnosis Using Ensemble Learning. Association for Computational Linguistics and Asian Federation of Natural Language Processing, pp.99-104.

[9] Zampieri, M., Tan, L. (2014). Grammatical Error Detection with Limited Training Data: The Case of Chinese. *Proceedings of the 1st Workshop on Natural Language Processing Techniques for Educational Applications (NLPTEA'14)*, pp. 69-74.

[10] Lee, L., Yu, L., Lee, K., Tseng, Y., Chang, L., & Chen, H. (2014). A Sentence Judgment System for Grammatical Error Detection. *COLING*, pp.67-70

[11] Gupta, A., (2014). Grammatical Error Detection and Correction Using Tagger Disagreement. Association for Computational Linguistics, pp. 49–51

[12] Yannakoudakis, H., Briscoe, T., Medlock., B., (2011). A new dataset and method for automatically grading ESOL texts. Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, pp.180–189.

[13] Tajiri, T., Komachi, M., Matsumoto, Y., (2012). Tense and Aspect Error Correction for ESL Learners Using Global Context. *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics* (Volume 2: Short Papers), pp.192-202.