DEPARTMENT OF PHILOSOPHY,
LINGUISTICS AND THEORY OF SCIENCE

MULTI-CLASS GRAMMATICAL ERROR DETECTION

Data, Benchmarks and Evaluation Metrics for the First Shared Task on Swedish L2 Data

Judit Casademont Moner

Master's Thesis:	30 credits
Programme:	Master's Programme in Language Technology
Level:	Advanced level
Semester and year:	Spring, 2022
Supervisor	Elena Volodina
Examiner	Lars Borin
Keywords	Grammatical Error Detection, L2 Swedish dataset, synthetic data, shared task

Abstract

Grammatical Error Detection (GED) is a challenging NLP task that has not received a lot of research attention in the recent years, especially in the Swedish language. However, in the world we live in, where there are more L2 (second language) learners than there have ever been, educational resources for students such as tools for grammar checking are needed. With this in mind, this Master's thesis presents the generation process of the Swedish MuClaGED (Multi-Class Grammatical Error Detection) dataset, which is going to be part of a Computational SLA (Second Language Acquisition) shared task and it will likely be useful for the future production of multilingual grammatical error detection systems. Once Swedish MuClaGED is obtained in this thesis, two main experiments are performed on it to test its capabilities and obtain baseline results in preparation for the aforementioned shared task. Moreover, this project also aims to tackle and explore the advantages, disadvantages and functionalities of the creation of hybrid error detection datasets by experimenting on producing GED models trained on the combination of original L2 learners' data with text corrupted with artificially generated syntactical errors.

Preface

First and foremost, I would like to thank my supervisor Elena Volodina for her guidance, support and patience during the writing of this thesis, and for stopping me from putting too much work on my shoulders. I also want to thank Arianna Masciolini for her advice during the process of generating the dataset, and Samir (Yousuf) Ali Mohammed for his suggestions and assistance during the process of experimentation with the models.

The main focus of this thesis, the generation of the Swedish MuClaGED (Multi Class Grammatical Error Detection) dataset[1], has been carried out in the context of the Computational SLA shared task on L2 data. The decisions regarding design and structure have been the result of discussion and collaboration with them. I want to thank the team behind it for allowing me to contribute with my work.

And obviously, thank you to my family and my friends back home, and to my classmates, although we have talked about this and they think it is a bit awkward to thank each other in the prefaces of our theses.

[1] The name of the dataset is provisional and subject to change.

Contents

1 Introduction . . . 1

1.1 Contributions . . . 2

1.2 Outline . . . 2

2 Background and Related work . . . 3

2.1 Grammatical Error Detection . . . 3

2.1.1 Grammatical Error Detection vs. Grammatical Error Correction . . . 4

2.1.2 Shared tasks . . . 5

2.2 Data for GED and GEC . . . 6

2.2.1 Obtaining data for GED and GEC . . . 6

2.2.2 Original learner data . . . 7

2.2.3 Synthetic datasets . . . 7

2.3 Evaluation metrics for GED . . . 8

3 Materials and methods . . . 9

3.1 Datasets . . . 9

3.1.1 SweLL-gold . . . 9

3.1.2 Synthetic dataset . . . 10

3.2 Format transformation of original learner data . . . 12

3.2.1 Token-level alignment . . . 12

3.2.2 Error label distribution . . . 13

3.2.3 Essay-to-sentences split . . . 14

3.2.4 Final information gathering . . . 14

3.3 Baseline modeling . . . 15

3.3.1 Swedish BERT sentence embeddings . . . 15

3.3.2 LSTMs and Bi-LSTMs . . . 15

3.3.3 Task 1: Error label detection on the sentence level . . . 16

3.3.4 Task 2: Error edit operation detection on the token level . . . 17

4 Results and analysis . . . 19

4.1 The final dataset(s) . . . 19

4.1.1 Possible limitations . . . 20

4.2 General results . . . 21
4.2.1 Best performing models for Task 1 . . . 22
4.2.2 Best performing models for Task 2 . . . 25
4.3 Performance by error type . . . 26
4.3.1 Performance by error type for Task 1 . . . 26
4.3.2 Performance by error type for Task 2 . . . 27
5 Discussion . . . 28
5.1 Discussion of the results . . . 28
5.2 Discussion of the dataset(s) . . . 29
6 Ethical considerations . . . 31
7 Conclusions . . . 32
7.1 The research questions answered . . . 32
7.2 Future work . . . 32
References . . . 34
8 Appendices . . . 37
8.1 Scores of the best performing model on original MuClaGED . . . 37
8.2 Scores of the best performing model on MuClaGED + 500 FakeDaLAJ (S-FinV) . . . 37

1 Introduction

Grammatical Error Detection (GED) and Grammatical Error Classification are two NLP tasks that have not been thoroughly explored in research, especially for the Swedish language, and even less for Second Language (L2) texts. Educational resources such as grammar checkers and evaluation tools for language students are currently in high demand, therefore investing time in investigation in the field of GED is crucial.

Many grammar checkers have been trained on correct text produced by native speakers. And previously, the focus on the field of Grammatical Error Detection has been on the language produced by non-native proficiency speakers in highly academic texts, such is the case of the Helping Our Own (HOO) shared task (Dale & Kilgarriff, 2011). Therefore, these systems do not work as well for Intelligent Computer-Assisted Language Learning (ICALL) systems. However, Leacock et al. (2014) have convincingly shown that foreign language learners' error correction will benefit from models trained on real L2 students' texts. Hence the importance of using authentic language learner data.

The purpose of this master's thesis is to prepare the MuClaGED dataset(s) for Swedish, obtain the baseline results and decide on the evaluation metrics for a multilingual Computational SLA (Second Language Acquisition) shared task that will be carried out by a team of researchers from various countries and on a set of various languages (English as well as other languages which are low-resource, such as Swedish, Czech, German and Italian). This will be the first time that original L2 (Second Language) learner data for the Swedish language will be used in a shared task. The goal of the shared task is to use L2 learners' texts to develop models capable of not only detecting grammatical errors (i.e., a binary classification between correct and incorrect), but also of working on multi-class error detection, that is classifying them into five main categories (Punctuation, Orthographic, Lexical, Morphological and Syntactic).

There are two main foci of this work:

- The generation of the dataset(s) necessary for the shared task in a CoNLL-like format where information is displayed on the token level; and
- The testing of two different tasks that can be performed on the generated dataset: Grammatical Error Detection and Classification on the sentence level by error types, and Grammatical Error Detection and Classification on the token level by the type of the necessary error correction operation (such as deletion, addittion or replacement).

At the same time, the possibility of enhancing original learner language datasets by using artificially generated data is explored in this thesis by testing models on datasets with different amounts of incorporated artificial error data.

In connection to these, three research questions are formulated:

- Is a dataset with this agreed structure appropriate for the shared task it is aimed at? If there are any challenges, should the task be redefined?
- Are error edit operations a viable approach for working with Grammatical Error Detection (GED)?
- Are the creation of synthetic error data (e.g., automatically generated) and the production of hybrid datasets a research path worth taking in the future?

The nature of this work is, thus, exploratory, where advantages and disadvantages of the proposed formats are studied, as well as data size needs in relation to the formulated shared tasks are evaluated.

1.1 Contributions

The primary contribution of this master's thesis is the creation of the MuClaGED dataset for Swedish containing 8,553 original L2 sentences coming from the SweLL dataset (Volodina et al., 2019) in a CoNLL format (Hajič et al., 2009), i.e., split by one token per line with three columns (for Addition, Deletion and Replacement error correction operations) displaying up to five possible error types (Punctuation, Orthographic, Lexical, Morphological and Syntactical).

A secondary contribution is the delivery of baseline results on the new L2 Swedish learner dataset to be provided to the participants in the Computational SLA shared task as a guide to compare their models against, as well as the establishment of the evaluation metrics to test the efficiency of the resulting GED models.

Additionally, insights on the possible viability of the incorporation of synthetic data, specifically FakeDaLAJ (S-FinV) (Casademont Moner & Volodina, 2022), to enhance the functionality and trainability of original learner datasets are provided.

1.2 Outline

In the next section, the theoretical background is introduced, with information on Grammatical Error Detection and Correction (GED and GEC) and the approaches on the field, shared tasks, the usage of original learner data as well as synthetic data and evaluation metrics for the shared tasks. Afterwards, the data and methods used for both the dataset generation and the error detection models are presented. Subsequently, an evaluation is performed. Next, the results of said evaluation are analysed, on the general and on the error level. Thereafter, a discussion is carried out, followed by the ethical considerations for this project. Finally, conclusions are drawn.

2 Background and Related work

In this section are presented the basic concepts of Grammatical Error Detection (GED), as well as some general approaches and models. Then, a brief overview of the closely related problem of Grammatical Error Correction (GEC) is also presented, and GED and GEC are compared. Then the existing most relevant tasks on the field are introduced. That is followed by a note on the necessary data, as well as on the possible data sources. And finally, an overview of the existing evaluation metrics for GED and GEC is made.

2.1 Grammatical Error Detection

Grammatical Error Detection (GED) is a challenging task in NLP and a field that has gained a lot of attention in the last few years. It is agreed on that, in the modern digital world, people tend to rely on a number of tools to learn new languages and improve their writing skills (Madi & Al-Khalifa, 2018b), as well as to assess their work (Rei & Yannakoudakis, 2016). The need for such tools exists in all languages, even in languages with a considerable research focus such as English, but especially in low-resource languages that are not researched as much, such as the case of L2 Swedish.

GED is the task of detecting grammatical errors in a written text (Yuan et al., 2021). It can be performed on the token level or on the sentence level. Traditionally, GED has been treated as a binary sequence labelling task where, for each token or sentence, a label of either 'correct' or 'incorrect' is assigned (Rei & Yannakoudakis, 2016). The task of GED can be extended into Grammatical Error Classification, where each of the errors needs to be labeled as belonging to one of the pre-established types. This task is also referred to as multi-class detection (Yuan et al., 2021).

Early approaches to GED focused on the detection of specific error types, such as prepositions or determiners. However, with time and with the help of shared tasks mainly focusing on GEC (See section 2.1.2), most of the current work attempts to detect all error types at once. As presented by Madi & Al-Khalifa (2018b), the main approaches that have been used to perform GED and GEC are rule-based, syntax-based and machine learning. Additionally, lately some techniques have explored the use of transformer-based models (Bryant et al., 2019).

In rule-based approaches, error patterns are manually predefined in the form of grammars or parsers that try to find a rule that matches a text (Madi & Al-Khalifa, 2018b). Although there are benefits to this technique, such as their ability to produce immediate feedback, their being easily understandable and the fact that new rules can be manually added, there are also disadvantages. The main disadvantage to this approach is the fact that the complexity of grammar requires considerable manual design of rules, which is time-consuming and costly.

In syntax-based techniques, a parser analyses the text and outputs a syntactic tree. If the parser is capable of producing a syntactic tree, the sentence is considered to be correct; otherwise, it is labeled as incorrect. In the 90s, parsers were based on computational grammars that were handwritten by linguists. With time, parsers were trained on large-scale annotated treebanks and became statistical-based (Leacock et al., 2014). An advantage is that, with this technique, all incorrect sentences captured in the computational grammar should be detected regardless of their nature. However, covering all syntax rules of a natural language is extremely complex and demanding (Madi & Al-Khalifa, 2018a).

The machine learning approaches can be divided into three subtypes: statistical-based methods, classification methods and deep learning models (Madi & Al-Khalifa, 2018a).

The statistical-based approaches use large annotated corpora to train models that work by assigning a prob-

ability score to a sequence of words to decide whether it contains an error or not, which is usually based on the frequency of appearance of each sequence in the corpus used (Madi & Al-Khalifa, 2018a). Although this approach allows for the creation of systems that are not language-dependent, a disadvantage of this approach is that it requires a notable amount of data to be able to generalize.

For the classification approaches, classifiers are trained on error-free text and make use of features (such as POS tags, grammatical relations, syntactic trees, etc.) to detect potential errors. The main advantage of this approach is that it is capable of detecting unseen errors. However, a disadvantage is that different classifiers need to be designed to tackle different error types, and certain features need to be manually written by experts (Madi & Al-Khalifa, 2018a).

The approach of using deep learning models, such as Neural Networks (NN) or Recurrent Neural Networks (RNN), has grown in popularity lately, as it does not require manually writing rules nor any other kind of feature engineering, as the model features are learned automatically. The methods in this area that have proven to be more effective in detecting and correcting grammatical errors are RNNs, such as Long Short-Term Memory (LSTM) (Madi & Al-Khalifa, 2018a).

Finally, with the recent arrival of transformers in the NLP field, transformer-based models have been explored as a new method to perform GED and GEC tasks. A recent example is that of Yuan et al. (2021), who have shown that transformer-based language models for multi-class GED for downstream GEC output considerably detailed results when detecting and classifying errors in written English. In their work, Yuan et al. (2021) prove that simply finetuning ELECTRA yields new state of the art results in multi-class error detection.

There is also the possibility of combining more than one of the aforementioned methods. Such is the case of Bell et al. (2019), who have used a bidirectional LSTM (Bi-LSTM) with contextual word embeddings from transformers (namely ELMo, BERT and Flair embeddings) to detect grammatical errors.

2.1.1 Grammatical Error Detection vs. Grammatical Error Correction

Grammatical Error Correction (GEC) is a task in NLP that has been getting a lot of attention recently. GEC systems are commonly used in text processors to check for spelling and grammar mistakes. They are also utilized to validate the output of machine translation systems as well as by second language learners and native speakers alike (Madi & Al-Khalifa, 2018a).

Whereas, as it has been mentioned in the previous section, GED is the task of detecting grammatical errors in a written text, GEC is the task of also finding a correction for these errors (Yuan et al., 2021). Oftentimes GED is regarded as a consequence of GEC, not as a task on its own (Rei & Yannakoudakis, 2016) because it is natural to think that, if one designs an algorithm capable of making corrections in a text, said algorithm also has the capacity to find the errors. However, it should not be assumed that all GEC systems, that can suggest corrections to grammatical errors, are at the same time capable of accurately detecting the presence of all error types (Rei & Yannakoudakis, 2016).

GEC is oftentimes treated as a monolingual sequence-to-sequence translation task, from grammatically incorrect input to grammatically correct output (Junczys-Dowmunt & Grundkiewicz, 2016), and GED is considered the task of labelling a sequence (either binarily or into error categories) (Yuan et al., 2021). GEC has also been treated as a classification or labelling task, where each sequence or token receives a label indicating the type of correction or edit operation that it should receive. This is the case of the work by Awasthi et al. (2019), which predicts edits instead of tokens.

All in all, GED and GEC can be considered complementary tasks, as correction cannot be fulfilled without

first detecting the errors. However, GED and grammatical error classification can be tasks on their own. Some studies have been using multi-class GED to improve the performance of GEC (Yannakoudakis et al., 2017) or to evaluate the output of GEC systems (Yuan et al., 2021). As stated by Rei & Yannakoudakis (2016), a system able to detect errors is relevant for the language learner even if it is unable to provide a correction for them.

2.1.2 Shared tasks

Shared tasks in the field of Grammatical Error Correction specifically began to appear in 2011, born from the lack of a common corpus and evaluation metrics to compare different systems with, which were preventing the progress in this area of study (Leacock et al., 2014). Several shared tasks have been carried out since then, which has allowed the sharing of annotated datasets, as well as the setting of common ground definitions, classifications of error types and evaluation methods.

The first one is the Helping Our Own (HOO) shared task that was carried out in 2011 (Dale & Kilgarriff, 2011) as a pilot task. The focus of the task was on errors and infelicities made by non-native speakers of English. They used fragments of texts collected from publications of the Association for Computational Linguistics which had been manually annotated with information on the correction of said errors and infelicities. Six teams took part in this task, and they had to detect, identify and correct errors distributed amongst thirteen different error types. Not all teams decided to focus on all error types, and the approaches used were very varied, from heuristic to data-driven approaches (Leacock et al., 2014).

The following year, in 2012, the Helping Our Own (HOO) shared task moved from academic language to text written by students taking official examinations in English (Dale et al., 2012). The data used came from the Cambridge Learner Corpus (CLC) (Yannakoudakis et al., 2011), which consisted of exam texts written by students undertaking the First Certificate in English (FCE) exams. The task focused on the detection and correction of errors involving prepositions and determiners instead of all grammatical errors in general. Fourteen teams participated in the shared task.

In 2013, the CoNLL-2013 shared task (Ng et al., 2013) continued to work with original data of students of English as a second language, and the seventeen participating teams had to perform grammatical error detection and correction on learner essays collected in the the NUCLE corpus, the NUS Corpus of Learner English (Dahlmeier et al., 2013). They only had to present the final corrected essay for evaluation. Three more error types were added to the ones present in the previous year's task, namely errors involving noun number, verb form and subject-verb agreement, so in total, the teams had to detect and correct five error types. The most used approaches by the teams were statistical classification, language modeling, machine translation and heuristic rules (Leacock et al., 2014).

The year after, the CoNLL-2014 shared task (Ng et al., 2014) encouraged the 45 participating teams to perform grammatical error correction on all error types, a total of 28 types, which increased the difficulty of the task compared to previous years' editions. The dataset used for this task was the NUS Corpus of Learner English (Dahlmeier et al., 2013), the same one that was used for the CoNLL-2013 shared task. It consists of 50 essays, with a total of 1,312 sentences. The most popular approaches to perform grammatical error correction on all error types were Language Model (LM)-based ones, and some teams used separate techniques to perform grammatical error correction and detection.

A few years after, the NLPCC 2018 Shared Task on grammatical error correction (Zhao et al., 2018) was organized. It was the first shared task for grammatical error detection and correction for Mandarin Chinese, focusing on linguistic errors made by learners of the language. 23 teams participated in this task, and the dataset was extracted from Lang-8, a webpage where learners of Chinese upload their essays to have them corrected by volunteer native speakers of the language. In this case, most participating teams treated error

correction as a Machine Translation (MT) task, and some other popular approaches were language models and rule-based models.

Finally, after five years since the CoNLL-2014 shared task, the Building Educational Applications (BEA) 2019 shared task (Bryant et al., 2019) was organized with the intention of comparing the progress made in the field since the previous shared task. The 24 participating teams had to perform grammatical error correction of all error types. They divided the task into three different 'tracks', depending on the amount of data that was used for training. The dataset used was the Write&Improve+LOCNESS corpus (Bryant et al., 2019). In this shared task, the most popular approaches were transformer-based models, machine translation models and convolutional neural networks (CNNs).

Finally, the Computational SLA shared task, to be carried out in the spring of 2023 and for which this thesis focuses on adapting the Swedish MuClaGed dataset, will aim at grammatical error detection and classification on a multilingual level.

2.2 Data for GED and GEC

Grammatical Error Detection and Correction (GED and GEC) are NLP tasks that require considerable amounts of high-quality annotated data (Kasewa et al., 2018), and such data is usually expensive to produce, or even problematic to obtain and gather. An alternative data approach to GED has also been to work on corpora of correct texts, with no grammatical errors.

There is a notable difference between datasets of native speaker language and those of language learners, and the first type is more abundant than the latter. The grammatical errors produced by native speakers of a language can oftentimes overlap with those made by the learners of said languages. However, language learners also make errors that appear rather infrequently in native speaker language (Leacock et al., 2014), hence the need for datasets handling learner errors specifically.

A significant challenge present in datasets for grammatical error detection and correction is data sparsity. Data for these tasks is found more unevenly in written language than for other tasks in the field of NLP, because the distribution of errors depends to a great extent on the learner's particularities, such as their level of proficiency, their native language(s), their age or the task at hand (Casademont Moner & Volodina, 2022). This data sparsity is considered to partially be the cause of the complexity in elaborating annotated corpora, not only because of the challenges posed by the need for annotators' agreement, but also because of how labour-intensive and expensive it is to perform manual annotation (Liu & Liu, 2017).

2.2.1 Obtaining data for GED and GEC

As mentioned in the previous section, obtaining data for the tasks of GED and GEC can be challenging, especially when the desired approach to tackle is in the group of statistical or machine learning methods, which require large quantities of labeled data. Written error data to perform GED and GEC especially for educational purposes can be obtained from two different types of sources: original learner data, namely texts written by L2 students, and synthetic data, which has been automatically generated. Whereas manually annotated human-made errors are representative and can therefore be useful to detect new errors, obtaining large datasets containing this kind of annotated data is expensive. And synthetic datasets are by some considered to deviate from the natural distribution of human-made errors (Yasunaga et al., 2021). Finding the perfect method to obtain high quality representative error data is still an ongoing and demanding challenge, and currently some datasets are formed by a combination of data sources (Leacock et al., 2014).

Labeled error datasets can be annotated on the sentence level or on the token level, although the latter is notably more common, habitually containing a diverse taxonomy of error types. Such is the case of the

Cambridge Learner Corpus First Certificate in English (CLC FCE) dataset by Yannakoudakis et al. (2011), with 77 error types. Another case is the data structure mentioned in the work by Bryant et al. (2017), where they present a taxonomy of 25 error types distributed amongst three edit operation categories, 'Unnecessary' (U), 'Missing' (M) and 'Replacement' (R).

2.2.2 Original learner data

The use of original learner data appears to be the most logical source for the creation of datasets that are used to train models to create systems and tools that are intended for L2 students to support them when learning a new language. To accurately perform GEC or GED on student-produced text, it is key to use data with a similar language use to that of the text we want to detect errors in (Leacock et al., 2014). Current corpora available to the public are usually extracted from formal and correct sources such as news sites or encyclopedias. The written texts' language style found in these corpora is different from the one used by students in their essays or other language learning tasks, usually. This means that it is possible that a language model trained on encyclopedic text will not perform accurately GED on L2 students' texts.

Collecting student-produced texts for GED and GEC datasets poses many advantages. However, as it has been previously mentioned, the collection of such data is costly, labour-intensive and time-consuming. Another aspect to take into consideration is the data protection policy, to preserve the anonymity of the students. This is why original learner data needs to go through an extra processing step called pseudonymization, where words disclosing any personal information (such as people's names or locations) are labeled in a generic manner to prevent anyone from recognizing the author (Volodina et al., 2019).

2.2.3 Synthetic datasets

It has been discussed in the recent years that the current lack of sufficient original learner data to train algorithms and models for the tasks of error detection, labeling and correction is pushing researchers in the field to explore the possibility of using synthetic datasets (that is, where the errors were not originally made by students, but artificially generated). As noted by Liu & Liu (2017), the amount of data required to train grammar checkers is considerable and researchers, unfortunately, have access to a very limited amount of annotated corpora. On top of that, the majority of annotated corpora are in English. Despite that, the need for more data exists in English, to say nothing of other, rather low-resource languages (Stahlberg & Kumar, 2021).

Recent studies have explored the generation of synthetic data, both as a way to enhance currently existing original datasets and as fully synthetic datasets. The approaches in this field can be grosso modo divided into two main groups: model-based approaches and rule-based approaches. Using model-based approaches has its advantages and disadvantages (Stahlberg & Kumar, 2021). On the one hand, using models to produce artificial error data allows the generated output to capture a wide variety of errors, and do so in a considerably short amount of time. The main problem with this approach, though, is that it requires the model to learn from a lot of data, and that is the initial problem itself. On the other hand, using rule-based approaches allows for the creation of synthetic datasets with little to no access to original error data (Grundkiewicz et al., 2019) as they are kept on an abstract level and can be designed based on the general rules of grammar. However, this approach is more time-consuming, as the rules need to be manually written. An example of this is the work by Casademont Moner & Volodina (2022), where the rules are designed based on the abstraction of patterns of part of speech (POS).

Overall, the use of synthetic datasets seems to be an area of research that has been gaining popularity in recent years, as the need for large amounts of data, especially for low-resource languages such as Swedish, increases. It is worth noting that the overall advantages are plenty. One does not need to take into consideration the student's data protection and privacy, unlike with datasets presenting original learner's data.

Additionally, the process of annotation is unnecessary, as the generation process itself is capable of automatically providing annotation for the errors it has produced. This field is still considerably new, and synthetic grammatical error datasets in Swedish are virtually non-existent.

2.3 Evaluation metrics for GED

The evaluation of both GED and GEC is a laborious task, since it is complicated to generalize and compare the performance of different models trained on different datasets. This is because the annotation of corpora usually follows different standards, and the evaluation of a model in this field is highly annotation-dependant (Xue & Hwa, 2014) (Leacock et al., 2014). Therefore, GED and GEC systems are oftentimes evaluated based on their performance overall, which can be ambiguous, as one model that is remarkably reliable in detecting or correcting a specific error type can perform badly overall when considering all error types, but a very robust system, despite it being notably specialized, is more desirable than a system that provides poorer output on a general level (Bryant et al., 2017). Therefore, an evaluation performed on the error level is relevant.

The inconsistency in the evaluation methods amongst different tasks and corpora inspired the organization of several shared tasks (See section 2.1.2), to be able to evaluate different systems using the same datasets and the same metrics, as the results of an evaluation cannot be transported and compared to those obtained from a different experimental settings (Evert, 2004).

Because GED consists of identifying whether a token or a sequence of tokens is correct or incorrect, traditional metrics generally used in NLP can be employed to evaluate the performance of GED. These metrics are accuracy, precision, recall and F-score. The overall performance metric is accuracy, where the total number of correct predictions is divided by the number of cases. Many researchers warn us that accuracy can sometimes be misleading for GED because, given an example where 10% of the tokens in a dataset are considered grammatical errors, a system that predicts everything as correct would get an accuracy of 90% (Leacock et al., 2014). Precision and recall are typically used as well, and they are computed by using counts of true positives (TPs), false positives (FPs) and false negatives (FNs). Recall divides TP over TP and FN, and precision divides TP by TP and FP (Leacock et al., 2014). Precision and recall are combined into the F-scores; F1 and F0.5. Most research nowadays computes F0.5 score instead of F1; both metrics are usually used when classes in a dataset are imbalanced. F1 is the harmonic mean of precision and recall, but in contemporary research a preference has arisen for F0.5 score, which is the weighted harmonic mean that gives more weight to precision than to recall in the calculation (Ng et al., 2014).

3 Materials and methods

This section firstly introduces the two datasets that were the source of the data for the generation of the new dataset(s) for the Computational SLA Shared Task for the Swedish language. Secondly, it is explained how they were transformed to fit the standards of the shared task, and the process is reported in detail. Finally, the data processing and the modeling to create the benchmark for the shared task are introduced.

3.1 Datasets

For this thesis, two datasets of grammatical errors in text written in the Swedish language are generated from two previous datasets. The two seed datasets are the original learner data SweLL-gold (Volodina et al., 2019) and FakeDaLAJ (S-FinV) (Casademont Moner & Volodina, 2022), a synthetic dataset of syntactical errors involving finite verbs. They both include error tagging and grammatical information on the token level to some extent.

The goal of the task is to reshape the two datasets so that they represent the error tagging information on the token level formatted in three error edit operation columns, **A**ddition, **D**eletion and **R**eplacement. Additionally, another goal is to incorporate part of the synthetic S-FinV dataset into the original learner dataset to explore the effectiveness of hybrid datasets, namely original learner data enriched with synthetic data.

3.1.1 SweLL-gold

The source of the data used to craft the corpus for this Master's thesis is the SweLL (Swedish Learner Language) gold corpus (Volodina et al., 2019). SweLL is an infrastructure with several collections, one of which is SweLL-gold, a collection of 502 essays (7,807 sentences in the source) written by learners of Swedish as a second language. The texts have been manually corrected and annotated by a team of researchers lead by Språkbanken, with the purpose to set the foundations of the research on Second Language Acquisition (SLA) on the Swedish language (Rudebeck & Sundberg, 2021).

The 502 texts in SweLL-gold have been written by adult (16+) second language learners of Swedish who are undergoing a formal education in Swedish as a Second Language, such as university, upper education courses or Swedish For Immigrants (SFI), or taking official examinations to test their knowledge of the language (Tisus or CEFR-based).

The learner texts have undergone a certain amount of manipulation, that is transcription, pseudonymization (to ensure the writers' anonymity) and normalization, and they are accompanied with demographic information of the writers and their performance in the form of metadata. The metadata includes, among others: age range, approximate level (one of "Avancerad", "Fortsättning" or "Nybörjare"[2] levels), course subject, date, education level, essay id, gender, grading scale and native language(s) of the learner. Not all parameters are provided for all the essays. For the transformation of the dataset, only approximate level, essay id and native language(s) have been kept, according to what it has been agreed with the Computational SLA shared task team in order to have all the datasets of the different languages in the same conditions.

The annotation consists of 29 error correction tags, which can be grouped into the following six subgroups: **P**unctuation, **O**rthographic, **L**exical, **M**orphological, **S**yntactical and Other. For this thesis, we consider the first five categories, namely POLMS. Furthermore, the Other category represents comments and tags

[2] "Advanced", "Continuing" (standing for "Intermediate") and "Beginner".

for unintelligible words in other languages, corrections that cannot be included in any of the established categories and preudonymization notes. Further information can be found in the annotation procedures (Rudebeck & Sundberg, 2021).

This dataset can be found in two file formats: xml and json. Each of the formats offers a slightly different representation of the data, at a different level of detail.

The xml format consists of two files: the source, which contains the original texts, and the target, which contains the corrected text. These files are split at the sentence level and the information is represented both on the essay level and on the token level (namely, each row in the file displays the information for one token). Firstly, on the essay level we can find metadata about the circumstances of the essay, such as the age range of the student, their approximate level of the language and their education level, the subject of the essay, their native language(s) and an essay ID. Secondly, the pieces of information for each token are many, but the most relevant are: dependency head, dependency relation, lemma, lexical information, POS tags, sense, universal morphological features and, most importantly, error tags. Although many tags can be seen in both files, some of them are found only in the source file, such as redundant tokens, whereas others are only visible in the target file, such as missing tokens. To be able to extract the error tag information from both files, they would need to be aligned, which they are not due to the sentence split.

The json format consists of one file that contains the source (original) essay, the target (corrected) essay and the ”edges”, which display information on alignment (namely, which tokens in the target text correspond to the tokens in the source one) as well as the error tags. In this format, the various sentences in one essay are treated as one unit (so, they have not been previously split), and no grammatical information on the token level nor metadata on the essay level (the only exception being the essay ID) are included.

3.1.2 Synthetic dataset

The source of the synthetic data, that is the error data that was not originally written by students of Swedish as a second language but generated automatically, is the FakeDaLAJ (S-FinV) dataset (Casademont Moner & Volodina, 2022). This dataset, released in connection to an accepted article at the 17th Workshop on Innovative Use of NLP for Building Educational Applications 2022, started as a master’s course project and is still a work in progress to experiment on the generation of synthetic error data to enhance performance of grammatical error correction and detection systems (GEC and GED). In this thesis it is utilized to test the use of synthetic datasets as a possible enhancer for the original dataset MuClaGED.

The FakeDaLAJ (S-FinV) dataset is the product of a pilot study on the generation of artificial error data by studying error patterns in authentic datasets, in this case the SweLL learner corpus (Volodina et al., 2019) and its one error per sentence version, the DaLAJ dataset (Volodina et al., 2021), which contains learner data modified to have each sentence include exclusively one error. The study shows one method for mimicking errors and replicating them on correct (namely, error-free) data, in this case on the COCTAILL corpus, which contains sentences from coursebooks of Swedish as a second language (Volodina et al., 2014). Figure 1 illustrates the generation process of the FakeDaLAJ (S-FinV) dataset. This work was started as a project for a course. However, the last three steps (namely initial evaluation, refinement and cleaning, and final evaluation, were carried out as part of this master’s thesis).

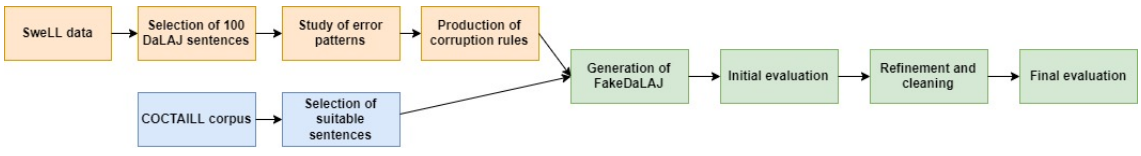

Figure 1: FakeDaLAJ (S-FinV) generation process.

This dataset contains errors of only the Syntactical type, specifically word order errors involving the placement of finite verbs in the sentence, tagged 'S-FinV' in the SweLL and DaLAJ datasets. It consists of 31,788 so-called "corrupted" sentences, where every sentence contains only one error. Therefore, one sentence might appear more than once in the dataset if the text allows for the inclusion of more than one error.

The errors of the type 'S-FinV' found in this dataset are formed from the combinations of a verb with four POS: verb in close relation with an adverb, verb in relation with a pronoun, verb in relation with a noun (and its modifiers, which can be adjectives, determiners or possessive pronouns), and verbs in relation with a proper noun. The four main so-called error patterns are distributed unevenly in the dataset (See Table 1), which represents the distribution of these patterns in the source DaLAJ data.

Error subtype	Produced errors
Pronoun-Verb	13,049
Adverb-Verb	9,922
Noun-Verb	8,041
Proper Name-Verb	776

Table 1: Error count by subtype of FakeDaLAJ (S-Finv).

The corruption pattern for errors regarding the position of verbs in relation to pronouns is PN-VB →VB-PN and VB-PN →PN-VB (where the first part is correct →the second is erroneous), as shown in the example below (Casademont Moner & Volodina, 2022).

> Jag **heter** Karin.[3] →**Heter** jag Karin.*
> English: My name is Karin.

The second corruption patterns, regarding the position of verbs in relation to adverbs, are: VB-AB →AB-VB* and AB-VB →VB-AB*, as it can be seen in the following example (Casademont Moner & Volodina, 2022).

> (...) om lillebror inte **ska** vara rädd för (...) →(...) om lillebror **ska** inte vara rädd för (...)*
> English: (...) if little brother must not be afraid of (...)

The main corruption pattern for errors involving verbs and nouns is VB-NN →NN-VB*, such as the example below. Since there is also the option for noun phrases, other patterns are: VB-DT-NN →DT-NN-VB, VB-PS-NN →PS-NN-VB and VB-JJ-NN →JJ-NN-VB (Casademont Moner & Volodina, 2022).

> Ibland **kommer** mormor. →Ibland mormor **kommer**.*
> English: Grandma comes sometimes.

Finally, the last corruption patterns, involving verbs in relation to proper nouns, are PM-VB →VB-PM and PM-PM-VB →VB-PM-PM. The example below is pseudonymized (Casademont Moner & Volodina, 2022).

[3] In the examples, the first sentence is correct and the second one contains one error. The verbs are in bold, whereas the parts of speech that are being treated are underlined.

Han visste inte om Brad Pitt **vann** priset. →Han visste inte om **vann** Brad Pitt priset.*

English: He didn't know if Brad Pitt won the prize.

3.2 Format transformation of original learner data

As it has been previously mentioned, the main contribution of this master's thesis is to transform the format of the existing original learner data in Swedish, the SweLL-gold dataset (Volodina et al., 2019), into the format agreed on by the Computational SLA shared task team to build Swedish MuClaGED.

The format specifications go as follows. The established taxonomy is of five error categories, to be distributed into three correction operation types, represented in columns. These error categories are the main five groupings found in the existing work by Volodina et al. (2019). The three error edit operation columns are inspired by the work by Bryant et al. (2017) but renamed slightly differently as ADR, standing for **A**ddition, **D**eletion and **R**eplacement. Additions are attached to the token after where the additional token (or tokens) should be added. For this purpose, a dummy token ('@') is added as the last position of the sentence, to contain the information in case a token needs to be added at the end of the sentence. As it can be seen in the example in Figure 2, each sentence in the dataset is formed of three lines containing the original text, the sentence id and the metadata respectively, followed by one line per token with the token index, the word itself and the three edit operation columns.

```
# text = Jag kunde inte forbi med de så dem var ensama . @
# sent id = 114
# metadata = Essay id = K48KT2, Approximate level = Fortsättning, L1
= Oromo, Split = Train
1   Jag      _   _       _
2   kunde    _   _       _
3   inte     _   _       _
4   forbi    _   _       ['L', 'S']
5   med      _   ['S']   _
6   de       _   _       ['M']
7   så       _   _       _
8   dem      _   _       ['M']
9   var      _   _       _
10  ensama   _   _       ['O']
11  .        _   _       _
12  @        _   _       _
```

Figure 2: Sentence example of the dataset. The columns, in order, are: token id, token, addition errors, deletion errors, replacement errors. Translation: 'I could not go by with them so they were alone.'

The text information for the dataset reformatting was gathered from the SweLL-gold dataset in the json format, which contains each full essay in two versions: its original form ('source') and its corrected form ('target'); and the token relation and error information ('edges').

To complete the format transformation of the dataset, a few steps have to be carried out. These steps involve creating a sentence-level alignment, simplifying the error tags and distributing them according to the ADR error correction operations, dividing the essays into sentences, and finally obtaining POS information and gathering metadata. The process is explained in the following sections.

3.2.1 Token-level alignment

To be able to distribute the error labels into the proper error correction operation type, represented in the ADR columns (see the previous section, Section 3.2), it is necessary to reach a linear alignment on the token-level between the original text ('source') and the corrected text ('target').

Two options for reaching token-level alignment were present initially. First, by using the xml format as a base, where we can find sentence-level splits that do not match at the sentence-id level (Source has 7,807 sentences, whereas Target has 8,137 sentences) due to error edits in the corrected file. Therefore, two alignment exercises were to be performed in this case: one where the text would become aligned on the sentence level and, after that, a second alignment on the token-level. This would have required more technical processes than investigating the second method, such as automatic sentence alignment techniques that are used to compare parallel documents, frequent in the field of machine translation (Moore, 2002). Consequently, the first alignment procedure option was discarded.

The second option, and the one that is used for the data transformation of this dataset, involves employing the json file, especially the 'edges', which capture the token-level relations. As it can be seen in the dummy example below, formatted as a table for visualization purposes (see Table 2), in the cases where one word is missing from the original (here, the word missing is the article 'ett', with the 'id': 't13') the 'edge' for the token is found with no matching token in the original (e.g., when a token in the source matches with one token in the target, it is shown in this manner : ids': ['s10', 't10'], whereas if it is not, it is shown differently: 'ids': ['t13']). Three edge types are found: one-to-one matches, where one token in the source corresponds to one token in the target; no matches, where a token or more than one token in the source cannot be matched to the target, or vice-versa; and matches on different number of tokens. In the last situation, the difference in the number of tokens is taken into account to determine whether it indicates an addition or a deletion.

Original text ('source')	**Corrected text ('target')**	**'Edges'**
{'id': 's10', 'text': 'Jag '}, {'id': 's11', 'text': 'bor '}, {'id': 's12', 'text': 'i '}, {'id': 's14', 'text': 'stor '}, {'id': 's15', 'text': 'huss '}	{'id': 't10', 'text': 'Jag '}, {'id': 't11', 'text': 'bor '}, {'id': 't12', 'text': 'i '}, {'id': 't13', 'text': 'ett '}, {'id': 't212', 'text': 'stort '}, {'id': 't213', 'text': 'hus '}	{'id': 'e-s10-t10', 'ids': ['s10', 't10'], 'labels': [], 'manual': False}, {'id': 'e-s11-t11', 'ids': ['s11', 't11'], 'labels': [], 'manual': False}, {'id': 'e-s12-t12', 'ids': ['s12', 't12'], 'labels': [], 'manual': False}, {'id': 'e-t13', 'ids': ['t13'], 'labels': ['M-Def'], 'manual': False}, {'id': 'e-s14-t212', 'ids': ['s14', 't212'], 'labels': ['M-Gend'], 'manual': False}, {'id': 'e-s15-t213', 'ids': ['s15', 't213'], 'labels': ['O'], 'manual': False}

Table 2: Slightly modified example from the SweLL dataset in the Json format.

With the information provided by the 'edges', a simple alignment can be reached on the token level. Whenever the number of ids from the source text is not equal to the number of ids from the target text, it can be considered that a misalignment is found. In consequence, an 'artificial' alignment is constructed in the form of a simple DataFrame where empty spaces or empty tokens are placed where additions or deletions take place. This way, blank lines indicate misalignments which, in turn, are a sign for either an addition or a deletion error.

3.2.2 Error label distribution

The error label distribution amongst the three ADR error correction operations is determined by either a strict manually-established limitation based on the linguistic analysis of the pattern of each error type, and an automatic distribution decided by the token-level extension of the error tag. Each error type tag has been looked into to observe its behaviour in the sentences, and it has been found that, whereas some error tags consistently behave in the same manner and only involve one of the three error edit operation types,

other error tags could be placed in more than one category. The 35 labels found in the SweLL taxonomy (Volodina et al., 2019) are reduced to five categories according to the first letter of the tag (e.g., 'M-Verb', which represents **M**orphological errors involving verbs and auxiliaries, becomes simply 'M'). The error label distribution, also referred to as 'label logic' can be seen in the table hereunder (See Table 3).

Correction operation categories	**Error labels**
Addition (A)	'S-M', 'S-Msubj', 'P-M'
Deletion (D)	'S-R', 'P-R'
Replacement (R)	'O', 'O-Cap', 'O-Comp', 'L-Der', 'L-FL', 'L-Ref', 'L-W', 'M-Adj/adv', 'M-Case', 'M-F', 'M-Gend', 'M-Other', 'M-Verb', 'S-Adv', 'S-Comp', 'S-FinV', 'S-Type', 'P-Sent', 'P-W'
No fixed category	'M-Def', 'M-Num', 'S-Clause', 'S-Ext', 'S-WO', 'S-Other'

Table 3: Error label distribution by error operation column.

Moreover, to be able to group the tokens that are part of the same error, which will be useful for future usage of the dataset for GED-related tasks, the error labels receive numbering whenever they belong to the same error. This is done by adding ':2' (and consecutively) to the error tag (See Figure 3).

```
# text = Mamma samma inställning har som vi .
# sent id = 2
# metadata = Essay id = synthetic, Approximate level = synthetic, L1 =
synthetic, Split = Train
1  Mamma         _  _  _
2  samma         _  _  ['S']
3  inställning   _  _  ['S:2']
4  har           _  _  ['S:3']
5  som           _  _  _
6  vi            _  _  _
7  .             _  _  _
```

Figure 3: Synthetic sentence example of the dataset. The columns, in order, are: token id, token, addition errors, deletion errors, replacement errors. Literal translation: 'Mom the same attitude has as us'.

3.2.3 Essay-to-sentences split

As the sentences are found grouped as essays in the SweLL dataset (Volodina et al., 2019), a sentence segmentation needs to be performed. Making use of UDPipe (Straka, 2018), which performs automatic splits of the input sentences, was attempted at first. However, hypothetically due to the lack of grammatical correction in a considerable number of sentences, UDPipe would output inaccurate sentences splits. Therefore, the more rudimentary technique of segmenting the essays into sentences by the means of punctuation placement was used. The quality of the sentence segmentation has not been systematically evaluated.

3.2.4 Final information gathering

The two final steps to generate the Swedish MuClaGED dataset are the addition of Part of Speech (POS) tags for the tokens, the gathering of essay metadata and providing an indication of the training, testing and development sets.

The POS tags are added to the dataset as a final column which is used merely for experimentation purposes in this master's thesis. However, this version of the dataset with extra token-level information will not be provided to the participants in the shared task. This task is done by parsing the sentences with the version of the Swedish UDPipe (Straka, 2018) parser trained on the Swedish-Talbanken treebank (Nivre et al., 2008) and attaching the obtained POS tag to the appropriate token in the dataset as an additional column.

Finally, the last step before the generation of the formatted dataset is the gathering of information to be

included in the metadata for each sentence. As commented earlier, the reformatting is carried out from the SweLL dataset in the json format, which includes no metadata besides the id of the essays. Therefore, the ids are used to extract the remaining necessary information from the xml file, namely the approximate level of the student and their native language(s).

Finally, a split suggestion of 80% for training, 10% for testing and 10% for development is provided in the form of one extra parameter in the metadata section.

3.3 Baseline modeling

Several machine learning experiments of slightly different design are performed on the generated datasets. The goal of these experiments is not to obtain the best-performing model possible, but to i. test the functionality of the corpora for some possible tasks in the field of GED, and ii. evaluate them to obtain tentative baseline results for the Computational SLA Shared Task, to compare the participants' scores.

Two tasks are performed with the data, making different use of the information found on the token level in the columns. The following sections illustrate the two different tasks and the models generated in detail. The first task is considered to be the main one for the definition of a baseline for the shared task, whereas the second task is conceived as an experiment to make use of the ADR correction operation distribution.

3.3.1 Swedish BERT sentence embeddings

Bidirectional Encoder Representations from Transformers, commonly referred to in the field of language technology as BERT, are large language representation models which provide pre-trained deep bidirectional representations for written text "by jointly conditioning on both left and right context in all layers" (Devlin et al., 2018). Through training, these models acquire substantial knowledge about how a language works. The model learns contextual relations amongst all words in a sequence of words and it produces rich feature representations (embeddings), both on the word level (the most frequently used for training models) and on the sentence level. BERT allows the building of models to perform a diverse amount of NLP tasks from its pre-trained word and sentence embeddings on a wide range of languages, and one of them is Swedish.

For the purpose of this master's thesis, it was decided to utilize sentence-level BERT embeddings instead of word-level ones due to the fact that incorrectly written words would be given split pre-trained embeddings by the model, which was trained on grammatically correct data. Therefore, the semantically meaningful sentence embeddings from KBLab's Swedish Sentence-BERT (Rekathati, 2021) are used on its 'cased' version.

3.3.2 LSTMs and Bi-LSTMs

Long Short-Term Memory (LSTM) and their bidirectional counterpart (Bi-LSTM) are a type of artificial neural networks called Recurrent Neural Networks (RNN). An RNN makes use of sequential data to feed the output of a previous step to the current training step (Abiodun et al., 2019). They are notable for their memory, which allows the output from previous steps to influence the following step. Nonetheless, RNNs do not learn well from long sequences of data (Sarker, 2021). To overcome this limitation involving the gradients of the neural network surged LSTMs and Bi-LSTMs.

LSTMs were first introduced in the work by Hochreiter & Schmidhuber (1997) with the aim of solving the problem regarding gradients. LSTMs are formed by various so-called gates, namely the 'Input gate', the 'Output gate' and the 'Forget gate' which manage the flow of information into the memory cell of the neural network, which is capable of storing data for long periods of time and long sequences of input. The 'Input

gate' is in charge of deciding which information enters the cell state, the 'Output gate' manages the outputs, and the 'Forget gate' determines what pieces of information are to be memorized.

Bi-LSTMs have the same structure as the original LSTM, but the difference between the two is that Bi-LSTMs are formed by two hidden layers (two LSTMs) that take in the input from opposite directions (i.e., one does take the input starting from the beginning and the other does it starting from the end), inputting data from the past and the future and consequently taking into consideration the entire context of an element in a sequence (Sarker, 2021), instead of just the previous elements, which is what is done by simple LSTMs. Bi-LSTMs are a common choice of method in many tasks involving context, which are most NLP tasks.

Since Bi-LSTMs consider the whole context of a token in a sequence and grammatical errors can oftentimes be related to other elements in the sentence, it seemed natural for the purpose of performing GED to make use of this type of RNN. For experimenting purposes, both LSTMs and Bi-LSTMs were employed in this thesis.

3.3.3 Task 1: Error label detection on the sentence level

Two GED tasks have been performed using the data. The first task has the aim of detecting the existence of errors on the sentence level and classifying them according to the five aforementioned categories (Lexical, Morphological, Orthographic, Punctuation and Syntactical). This is done regardless of the error frequency in the sentence, as the goal is simply to detect the existence of some error of a certain type. For example, for an input sentence that contains two Orthographic (O) errors and three Syntactic (S) errors, the correct output for the model would be to predict the tags 'O' and 'S'.

To perform this task, two distinct word embedding methods have been utilized. In the first method, they were created from a simple mapping from words to real numbers through a 'nn.Embedding' layer, which generates a M x N matrix, where M is the number of words in the vocabulary and N is the size of each word vector. The second method utilized Swedish BERT sentence embeddings extracted using the pretrained Swedish sentence transformer (Rekathati, 2021).

The models constructed are either Long Short-Term Memory (LSTM) or Bidirectional Long Short-Term Memory (Bi-LSTM) neural networks. The Bi-LSTM has the same structure as the LSTM, but the difference is that it adds one more LSTM layer, which looks at the input information in reverse.

For each model type in the lists, an average of five models with different hyperparameters have been trained in order to find the best performing one in each of the evaluation parameters.

3.3.3.1 Models

The models that were trained for this first task can be divided into two general groupings depending on the nature of the data they were trained on: whether it is fully original learner data or learner data mixed with different amounts of synthetic data.

The models trained purely on original learner data are of the six types hereunder:

1. Simple LSTM with word embeddings generated automatically from the Embedding dimension in the model (LSTM model 1, LSTM model 2, LSTM model 7, LSTM model 8).
2. Simple LSTM with word and POS embeddings generated automatically from the Embedding dimension in the model (LSTM model 1, LSTM model 2, LSTM model 7, LSTM model 8).
3. Bi-LSTM with word embeddings generated automatically from the Embedding dimension in the

model (Bi-LSTM model 3, Bi-LSTM model 4, Bi-LSTM model 5, Bi-LSTM model 6).
4. Bi-LSTM with word and POS embeddings generated automatically from the Embedding dimension in the model (Bi-LSTM model 3, Bi-LSTM model 4, Bi-LSTM model 5 and Bi-LSTM model 6).
5. Simple LSTM with sentence embeddings from Swedish BERT (BERT LSTM 3, BERT LSTM 4, BERT LSTM 6).
6. Bi-LSTM with sentence embeddings from Swedish BERT (BERT Bi-LSTM 1, BERT Bi-LSTM 2, BERT Bi-LSTM 5).

The models trained on original learner data with incorporated synthetic data from FakeDaLAJ (S-FinV) (500, 1,000 or 1,500 synthetic sentences) are of the following types:

1. Simple LSTM with word embeddings generated automatically from the Embedding dimension in the model (for the MuClaGED + 500 FakeDaLAJ: LSTM model 1, LSTM model 3, LSTM model 5, LSTM model 7; for the MuClaGED + 1000 FakeDaLAJ: LSTM model 1, LSTM model 3, LSTM model 5, LSTM model 7; for the MuClaGED + 1500 FakeDaLAJ: LSTM model 1, LSTM model 3, LSTM model 5, LSTM model 7).
2. Bi-LSTM with word embeddings generated automatically from the Embedding dimension in the model (for the MuClaGED + 500 FakeDaLAJ: Bi-LSTM model 2, Bi-LSTM model 4, Bi-LSTM model 6, Bi-LSTM model 8; for the MuClaGED + 1000 FakeDaLAJ: Bi-LSTM model 2, Bi-LSTM model 4, Bi-LSTM model 6, Bi-LSTM model 8; for the MuClaGED + 1500 FakeDaLAJ: Bi-LSTM model 2, Bi-LSTM model 4, Bi-LSTM model 6, Bi-LSTM model 8).
3. Simple LSTM with sentence embeddings from Swedish BERT (for the MuClaGED + 500 FakeDaLAJ: BERT LSTM model 3, BERT LSTM model 4, BERT LSTM model 6; for the MuClaGED + 1000 FakeDaLAJ: BERT LSTM model 3, BERT LSTM model 4, BERT LSTM model 6; for the MuClaGED + 1500 FakeDaLAJ: BERT LSTM model 3, BERT LSTM model 4, BERT LSTM model 6).
4. Bi-LSTM with sentence embeddings from Swedish BERT (for the MuClaGED + 500 FakeDaLAJ: BERT Bi-LSTM model 1, BERT Bi-LSTM model 2, BERT Bi-LSTM model 5; for the MuClaGED + 1000 FakeDaLAJ:BERT Bi-LSTM model 1, BERT Bi-LSTM model 2, BERT Bi-LSTM model 5; for the MuClaGED + 1500 FakeDaLAJ: BERT Bi-LSTM model 1, BERT Bi-LSTM model 2, BERT Bi-LSTM model 5).

3.3.4 Task 2: Error edit operation detection on the token level

The second GED task combines binary error detection and classification with the error operation prediction (Addition, Deletion and Replacement). The goal is to detect the presence of an error focusing on the necessary operation to correct the error, regardless of the error type. That is, for a token-level annotation that states that a certain token contains a Syntactic error of the Addition (A) edit operation type and a Morphological error of the Replacement (R) operation type, the model for this task should predict 'A' and 'R'.

To perform this task, two different sequence lengths (n-grams) were fed to the models: trigrams (i.e. the token accompanied with the previous and the following tokens) and 5-grams (i.e. the token accompanied with the previous two and the following two tokens). In the cases where a token is at the beginning or at the end of the sentence, dummy tokens are provided in the form of padding (namely '[START]' and '[END]').

3.3.4.1 Models

The models trained for the second task have been all trained on fully original learner data only, and they can be divided into two general categories depending on the n-gram length:

1. Trigram models trained on purely original learner data.
2. Five-gram models trained purely on original learner data.

Each of the aforementioned model types has been modeled following two methods:

1. Simple LSTM with word embeddings generated automatically from the Embedding dimension in the model (for the trigrams: LSTM model 1, LSTM model 2, LSTM model 3, LSTM model 4; for the five-grams: LSTM model 1, LSTM model 2, LSTM model 3, LSTM model 4).
2. Bi-LSTM with word embeddings generated automatically from the Embedding dimension in the model (for the trigrams: Bi-LSTM model 1, Bi-LSTM model 2, Bi-LSTM model 3; for the five-grams: Bi-LSTM model 1, Bi-LSTM model 2, Bi-LSTM model 3).

For each of the four groups and the two model types, which result in a total of eight model types, four models were trained on a variation of parameters with the goal of obtaining the best performing one.

4 Results and analysis

In this section, the information about the resulting dataset is gathered, and some possible limitations are noted. After that, the general results are presented for both Task 1 and Task 2. For Task 1, the best performing models trained on each data type are defined and compared. For task two, the models with better results trained on two n-gram sizes are explained and the results are compared. Finally, the results are analyzed on the error type level for both tasks.

4.1 The final dataset(s)

The final Swedish MuClaGED dataset, based entirely on the SweLL-gold dataset (Volodina et al., 2019), is formed by a total of 8,553 sentences (155,415 tokens). These sentences are represented in a 'CoNLL-like' format, where each sentence is formed by: an initial line with the full text and a second line with metadata followed by one line for each individual token in the sentence. The metadata contains information about the original essay id, the approximate level of the student, their native language or languages and the split the sentence belongs to (either train, test or dev splits). The token-level information consists of three error correction operation categories, namely ADR, standing for **A**ddition, **D**eletion and **R**eplacement. See the example below (Figure 4):

```
# text = Du får desutom . @
# sent id = 5101
# metadata = Essay id = C81CT14, Approximate level = Nybörjare, L1 =
Dari, Persiska, Split = Train
1  Du       _  _      _
2  får      _  _      _
3  desutom  _  _      ['O']
4  .        _  ['P']  _
5  @        _  ['P']  _
```

Figure 4: Sentence example from the Swedish MuClaGED dataset. The columns, in order, are: token id, token, addition errors, deletion errors, replacement errors. Translation: 'You receive in addition.'

From this initial version of the dataset, two other versions have been generated, which include more fine-grained detail on the token-level information in the form of two extra columns. The second version includes an extra column with the full error label as it was presented in the SweLL corpus as extra information; for example, the label 'M-Adj/adv', which is labeled as simply 'M' in the Replacement column, will appear in the fourth column, namely the 'Full label' column. Finally, the third version includes one additional column with the POS of each token. For the purpose of the Computational SLA shared task, only the first dataset with the least quantity of information is to be released.

In the dataset we can find the following error distribution by token (See Table 4). This means that one sentence might contain more than one error of the same type, and one token might be involved in more than one type of error at once.

Error type	**Number of tokens containing this error**
Lexical	4,862
Morphological	7,957
Orthographic	4,360
Punctuation	2,888
Syntactical	7,422
Total count of errors	27,489
Error-free tokens	127,926

Table 4: Error distribution by token

The following table (See Table 5) presents the error distribution on the sentence level, that is, it shows how many sentences contain at least one error for each of the types. This means that, even though one sentence might contain, for example, 3 grammatical errors of the type 'Syntactic', it is only counted once.

Error type	Number of sentences containing this error
Fully correct sentences	2,100
Lexical	3,146
Morphological	3,922
Orthographic	2,688
Punctuation	1,843
Syntactical	3,763

Table 5: Error distribution by sentence

The table hereunder shows the error distribution by correction operation categories, whether they are of the Addition, Deletion or Replacement types (See Table 6).

Column type	Number of errors
Addition (A)	6,120
Deletion (D)	2,394
Replacement (R)	20,058

Table 6: Error counts by column

Finally, one can also take into consideration the number of sentences representing each of the three proficiency levels (see Table 7). It is notable to mention that there are as many sentences belonging to students with an advanced knowledge of the language as there are sentences that were written by beginner and intermediate level students.

Proficiency level	Number of sentences
Nybörjare (Beginner)	3,280
Fortsättning (Intermediate)	1,012
Avancerad (Advanced)	4,261

Table 7: Sentence counts by proficiency level

Several other sub-datasets have been generated by adding different amounts of synthetic data from the experimental FakeDaLAJ (S-FinV) synthetic dataset. The following are the currently existing versions (each version adds the specified number of sentences to the Syntactic categories in Table 4 and Table 5 and to the Replacement category in Table 6): MuClaGED + 250 FakeDaLAJ, MuClaGED + 500 FakeDaLAJ, MuClaGED + 1000 FakeDaLAJ, MuClaGED + 1500 FakeDaLAJ, MuClaGED + 2000 FakeDaLAJ, MuClaGED + 3000 FakeDaLAJ, MuClaGED + 4000 FakeDaLAJ.

4.1.1 Possible limitations

The aforementioned datasets have a few limitations in regard to their potential use in the fields of GED and GEC.

Firstly, as a consequence of the nature of the split regarding the sentence-level alignment, the original sentences are not accompanied with their corrected version. This is due to the fact that there are cases where punctuation errors separate sentences, and this provokes misalignments, which causes some of the sentences not to have an equivalent pair. Therefore, this dataset cannot be used for correction purposes, as no correction is provided in the form of metadata. An example can be seen below (see Figure 5), where the original is one sentence, but the corrected text would be two sentences as there is a missing punctuation ('P-M') error, marked as simply 'P', of the Addition type, located in the third column.

```
# text = Tre barn har jag Jag tyckem min lägenhet . @
# sent id = 103
# metadata = Essay id = G58GT4, Approximate level = Nybörjare, L1 =
Arabiska, Kurdiska, Split = Test
1   Tre       _      _  _
2   barn      _      _  _
3   har       _      _  _
4   jag       _      _
5   Jag       ['P']  _  _
6   tyckem    _      _  ['O']
7   min       ['S']  _  _
8   lägenhet  _      _  _
9   .         _      _  _
10  @         _      _  _
```

Figure 5: Sentence example with a missing punctuation error (on token 5). The columns, in order, are: token id, token, addition errors, deletion errors, replacement errors. Translation: 'I have three children I like my apartment.'

A second limitation is the size of the dataset. There is the possibility that, for certain machine learning purposes which require considerable amounts of data to properly train models, the quantity of data present in the main dataset, of 8,553 sentences, might not suffice.

A third possible inconvenience is the imbalance in the amount of error labels, namely the fact that the five possible errors are not found in the same frequency amongst the sentences forming the dataset. It is therefore likely that the models trained with this data will have a better performance on the most frequent error types (Syntactical and Morphological).

Additionally, the datasets that increase the number of available sentences with the synthetic data only do so by simultaneously adding to the Syntactic error type and the Replacement column. This expands the dataset only on one aspect, which might make the resulting models become specialized in the detection of one specific error type. Another consideration is that the FakeDaLAJ (S-FinV) synthetic dataset presents only one grammatical error per sentence, whereas many of the sentences in the original learner data are likely to present more than one.

A secondary limitation to the synthetic additions to the dataset is that the metadata available does not provide information that could be useful for sociological statistical studies as the original data could. For example, with the original learner data one could explore the most frequent errors committed by native speakers of a certain language, or by learners with a certain proficiency level of the language they are learning. However, as the metadata for Approximate level and L1 is set to 'Synthetic', studies that would dissect the language used by certain groups of students would not be possible.

4.2 General results

The evaluation metrics chosen to test the performance of the models were the traditional main metrics employed in NLP, namely accuracy, precision, recall, F1-score and F0.5-score. These were performed on all error labels overall as well as individually to be able to find the best performing models for each of them. Additionally, the instances where the model would predict all five error tags correctly for one sentence were counted and averaged.

The model predictions are of a decimal number between 0 and 1 for each of the error tags (in the shape of [Orthographic, Lexical, Morphological, Punctuation, Syntactic]) and, to determine the real binary score, it is rounded up or down to the closest full number. That is, a probability of ≥ 0.5 will be considered a 1

(standing for True, the existence of a tag of a certain type), and a probability of < 0.5 will be considered a 0 (standing for False, the non-existence of a tag of a certain type).

Accuracy, precision, recall and F1-score are calculated on the final resulting binary scores, collected after each iteration of the testing data. The functions used are Scikit-learn's accuracy_score, recall_score, precision_score and f1_score. Finally, from the results of the recall and the precision, F0.5 score is calculated following the formula:

$$F0.5 = \frac{1.25 \cdot Precision \cdot Recall}{0.5 \cdot Precision + Recall}$$

4.2.1 Best performing models for Task 1

It has been decided that the best performing models are to be defined by the F0.5 score, as it is most commonly done in GED tasks currently (See Section 2.3), as well as the percentage of fully correct label predictions. That is, the results for accuracy, precision, recall and F1 are not considered when determining the best performing model of each type. However, as the F0.5 score is the weighted harmonic mean of precision and recall, these two measures are indirectly taken into account. The F0.5 score is computed by considering each individual prediction in the label of five error categories (Lexical, Morphological, Orthography, Punctuation and Syntactical, in that order) for each sentence. On the other hand, the correct full label, in this case, refers to the percentage of instances when the totality of the five error categories is predicted correctly on the sentence level.

Hereunder are presented the best performing models for task one, Error label detection on the sentence level. They are grouped by the dataset that they were trained on and, for each dataset, the best result for each modeling technique is presented. The best performing model overall is reported as well.

4.2.1.1 Fully original learner data

The best performing models using fully original data and automatically generated embeddings via the Embedding dimension are LSTMs (LSTM models 1 and 7), whereas the models employing BERT sentence-level embeddings perform better when built through a Bidirectional LSTM (BERT model 5).

The LSTM model 1 trained for 3 epochs, with an embedding dimension of 1024 and a hidden dimension of 64. The output of this model, as well as of all of the models generated for this task, is of 5. This model produces the highest number of fully correct label predictions.

The LSTM model 7 trained for 25 epochs, with an embedding dimension of 256 and a hidden dimension of 64. This model obtains the highest F0.5 score of its embedding type class.

The BERT model 5 trained for 25 epochs, with an embedding dimension of 768 and a hidden dimension of 64. This model obtains the highest F0.5 of all the models trained on this amount of data.

Hereunder (See Table 8) are the general results for the three best performing models. The higher results for each category are highlighted.

4.2.1.2 Fully original learner data using POS information

	Simple Embeddings		BERT
	LSTM model 1	LSTM model 7	Bi-LSTM model 5
Correct full label	25.14%	19.97%	25.73%
F0.5	0.22351797	0.47320026	0.58780357

Table 8: Best performing models for Task 1 on fully original data.

The two best performing models using fully original data, automatically generated embeddings via the Embedding dimension added element-wise (i.e., employing the method 'torch.add') to the extra grammatical information provided by the POS tags are a simple LSTM (LSTM model 7) and a Bidirectional LSTM (Bi-LSTM model 3). The performance of these models is slightly lower than that of the model employing BERT embeddings.

The LSTM model 7 trained for 5 epochs, with an embedding demension of 256 and a hidden dimension of 128. This model obtains the highest percentage of fully correct label predictions.

The Bi-LSTM model 3 trained for 5 epochs as well, with an embedding dimension of 1024 and a hidden dimension of 64. This model's F0.5 score is the highest of its class.

The following table (See Table 9) presents the general results for the best performing models employing POS tags information.

	Simple embeddings with POS information	
	LSTM model 7	Bi-LSTM model 3
Correct full label	18.71%	14.15%
F0.5	0.43428658	0.47459491

Table 9: Best performing models for Task 1 on fully original data employing extra information provided by POS.

4.2.1.3 Original learner data with 500 synthetic sentences

For models employing original learner data augmented with 500 synthetic sentences from the FakeDaLAJ (S-FinV) dataset, the four best performing models are a simple LSTM (LSTM model 7) and Bidirectional LSTM (Bi-LSTM model 6) with automatically generated word embeddings, and two Bidirectional LSTMs employing BERT embeddings (Bi-LSTM model 2 and 5).

The LSTM model 7 trained for 5 epochs, with an embedding dimension of 1024 and a hidden dimension of 64. This model's percentage of fully correct label predictions is the highest of its class, making use of automatically generated embeddings.

The Bi-LSTM model 6 trained for 5 epochs as well, with an embedding dimension of 256 and a hidden dimension of 64. This model has the highest F0.5 score amongst all models of this class.

The Bi-LSTM model 2 using BERT sentence-level embeddings trained for 25 epochs, with an embedding dimension of 768 and a hidden dimension of 64. This model has the highest correct full label prediction percentage of all models involving the original dataset enhanced with 500 FakeDaLAJ (S-FinV) sentences.

The Bi-LSTM model 5 with BERT embeddings has the same parameters as the Bi-LSTM model 2 with a dropout of 0.1 instead of 0.2. Therefore, their results are notably similar.

In the following table (See Table 10) are shown the scores for the aforementioned models.

	Simple embeddings		BERT	
	LSTM model 7	Bi-LSTM model 6	Bi-LSTM model 2	Bi-LSTM model 5
Correct full label	18.85%	8.87%	31.59%	30.82%
F0.5	0.24912726	0.40913820	0.62620097	0.63916364

Table 10: Best performing models for Task 1 on MuClaGED + 500 sentences from FakeDaLAJ (S-FinV).

4.2.1.4 Original learner data with 1,000 synthetic sentences

The three best performing models trained on original learner data augmented with 1000 synthetic sentences from the FakeDaLAJ (S-FinV) dataset are: a simple LSTMS employing automatically generated word embeddings (LSTM model 1) and a Bidirectional LSTM (Bi-LSTM model 8) and a Bidirectional LSTMs using BERT embeddings (Bi-LSTM model 1).

The LSTM model 1 trained for 25 epochs, with an embedding dimension of 256 and a hidden dimension of 64. This model has the highest F0.5 score out of the models with automatically generated embeddings for this dataset.

The Bi-LSTM model 8 trained for 5 epochs, with an embedding dimension of 1024 and a hidden dimension of 64. This model provides the highest percentage of full label correct predictions of this type.

The BERT Bi-LSTM model 1 trained for 5 epochs, with an embedding dimension of 768 and a hidden dimension of 64. This is the best performing model using BERT embeddings, as well as the best model overall trained on this dataset.

The table hereunder (See Table 11) presents the scores for each of the models in this section.

	Simple embeddings		BERT
	LSTM model 1	Bi-LSTM model 8	Bi-LSTM model 1
Correct full label	6.81%	9.74%	30.89%
F0.5	0.43580971	0.35967741	0.57187017

Table 11: Best performing models for Task 1 on original data enhanced with 1,000 sentences from FakeDaLAJ (S-FinV).

4.2.1.5 Original learner data with 1,500 synthetic sentences

For models trained on a dataset of original learner data enhanced with 1,500 synthetic error sentences, the best performing model trained with automatically generated word embeddings is a simple LSTMs (LSTM model 5), whereas the best performing models trained with BERT embeddings are one simple LSTM (LSTM model 3) and one Bidirectional (Bi-LSTM model 5).

The simple LSTM model 5 trained for 5 epochs, with an embedding dimension of 256 and a hidden dimension of 64. This is the model reporting a higher percentage of full error labels guessed correctly as well as the highest F0.5 score.

The BERT LSTM model 3 trained for 25 epochs, with an embedding dimension of 768 and a hidden dimension of 64. This model presents the higher percentage of full error labels predicted correctly out of the BERT models as well as the best for all of the models trained with this dataset overall.

The BERT Bi-LSTM model 5 trained with the same parameters as model 3, but the difference was that it is trained on a slightly lower dropout of 0.1 instead of 0.2. This model achievs the highest F0.5 scores out

of all the models for this dataset.

The following table (See Table 12) illustrates the models' performance.

	Simple embeddings	**BERT**	
	LSTM model 5	LSTM model 3	Bi-LSTM model 5
Correct full label	13.83%	34.83%	33.33%
F0.5	0.45776051	0.59689787	0.59810412

Table 12: Best performing models for Task 1 on original data enhanced with 1,500 sentences from FakeDaLAJ (S-FinV).

4.2.1.6 Cross-dataset comparison

Below is a table (See Table 13) comparing the best scoring models for task 1 across all datasets (fully original learner data and the different combinations of MuClaGED with FakeDalaj (S-FinV)). Only the models yielding the highest scores for each dataset combination are included. The highest percentage of correct full labels on the sentence level and F0.5 score amongst all models are highlighted.

Data type	**Model name**	**Correct full label**	**F0.5 score**
Original learner data	BERT Bi-LSTM model 5	25.73%	0.58780357
Original learner data with POS	LSTM model 7	18.71%	0.43428658
	Bi-LSTM model 3	14.15%	0.47459491
Original learner data + 500 FakeDaLAJ	BERT Bi-LSTM model 2	31.59%	0.62620097
	BERT Bi-LSTM model 5	30.82%	0.63916364
Original learner data + 1000 FakeDaLAJ	BERT Bi-LSTM model 1	30.89%	0.57187017
Original learner data + 1500 FakeDaLAJ	BERT LSTM model 3	34.83%	0.59689787
	BERT Bi-LSTM model 5	33.33%	0.59810412

Table 13: Best performing models for Task 1 comparison.

The complete obtained scores of the best performing model in terms of F0.5 scores on fully original learner data (namely BERT Bi-LSTM model 5) and on the hybrid original learner data with synthetic data (namely BERT Bi-LSTM model 5 for MuClaGED + 500 FakeDaLAJ sentences) are displayed fully in Appendix 8.1 and Appendix 8.2 respectively.

4.2.2 Best performing models for Task 2

Similarly to the criterion established to determine the best performing models for Task 1 (See Section 4.2.1), the F0.5 score is taken as the main measure. Additionally, the number of full labels predicted correctly (i.e., the three labels representing the ADR columns) is taken into account. Accuracy, precision, recall and F1 score, although measured, are not taken into consideration to decide the best performing model.

In the following two sections are presented the best performing model for task two, Error column detection on the token level. They are grouped by the number of neighbouring tokens shown to the model to predict the label, namely trigrams and five-grams.

4.2.2.1 Trigram models on fully original learner data

For models trained on original learner data to detect the belonging of an error tag in the addition, deletion and replacement categories based on the information provided by trigrams, the model that yield the best results is a simple LSTM (LSTM model 1).

The LSTM model 1 trained for 3 epochs, with an embedding dimension of 256 and a hidden dimension of 64, and presents the best results on both number of fully correct predictions and F0.5 score. It is illustrated in the table below (See Table 14).

	Simple embeddings
	LSTM model 1
Correct full label	76.17%
F0.5	0.11168091

Table 14: Best performing trigram models for Task 2 on original data.

4.2.2.2 Five-gram models on fully original learner data

For models trained fully on the original learner dataset Swedish MuClaGED based on five-grams, the best performing ones were, the two best performing ones are two simple LSTMs (LSTM model 1 and 4).

The LSTM model 1 trained for 3 epochs, with an embedding dimension of 256 and a hidden dimension of 64. This model obtains the highest percentage of correct full label predictions for this task.

The LSTM model 4 trained for 40 epochs, with an embedding dimension of 1024 and a hidden dimension of 128. It yields the best F0.5 score for this task with the five-grams.

The following table (See Table 15) presents the aforementioned models with their scores.

	Simple embeddings	
	LSTM model 1	LSTM model 4
Correct full label	80.75%	76.13%
F0.5	0.06613478	0.10322801

Table 15: Best performing five-gram models for Task 2 on original data.

4.3 Performance by error type

In this subsection, the scores of the best performing models are compared on the error type level. For Task 1, that is on the five label taxonomy established in Section 3.2.2, namely Lexical, Morphological, Orthographic, Punctuation and Syntactical. For Task 2, the three types, as established in Section 3.2, are Addition, Deletion and Replacement. Similarly to the general results, the F0.5 score is considered to be the determining factor.

4.3.1 Performance by error type for Task 1

Models trained on all dataset combinations for Task 1, Error label detection on the sentence level, are compared in the table below (See Table 16). This is done to be able to see if there is a certain amount of synthetic data that enhances the performance of a certain error type.

The Lexical label involves errors regarding word formation, foreign language words, anaphoric expressions and wrong word or phrase. The Morphological label involves errors regarding word forms: the choice of case, definiteness of constructions, declension and conjugation, gender, number and inflection in general.

The Orthographic label involves errors that have to do with spelling mistakes, upper and lowercase letters, spacing and hyphening. The Punctuation label signals errors involving the addition, removal or modification of punctuation and the splitting of sentences. And the Syntactical label comprises errors involving the placement of words in a sentence (adverbs, verbs, etc.), clause structure, restructuring of multi-word expressions and missing clauses. (Rudebeck & Sundberg, 2021).

Data type	Model name	Lexical	Morphological	Orthographic	Punctuation	Syntactical
Original learner data	BERT Bi-LSTM model 5	0.54894179	0.60539215	0.57565789	0.46072507	0.64680232
Original learner data with POS	BERT Bi-LSTM model 3	0.44901065	0.51724137	0.33475783	0.05263157	0.55408472
Original learner data + 500 FakeDaLAJ	BERT Bi-LSTM model 5	0.60634328	0.63834422	0.61026936	0.56034482	0.69732297
Original learner data + 1,000 FakeDaLAJ	BERT Bi-LSTM model 1	0.43002915	0.58045554	0.496453900	0.25390625	0.67343173
Original learner data + 1,500 FakeDaLAJ	BERT Bi-LSTM model 5	0.51798561	0.58823529	0.50641940	0.37499999	0.71934945

Table 16: Best performing models of all data types for Task 1 by error type.

4.3.2 Performance by error type for Task 2

Models trained to detect the classification of error labels into the three ADR categories based on trigrams and five-grams are compared in the table hereunder (See Table 17). Although having obtained high accuracies, the F0.5 scores are notably low for this tasks.

The Addition category involves error types whose correction operation would imply the addition of one token or more, whereas labels placed under Deletion represent errors that should be corrected with the deletion of a token. The Replacement category gathers errors where the token itself needs to be modified in order to be correct.

N-gram	Model name	Addition	Deletion	Replacement
Trigram model	LSTM model 1	0.02515090	nan	0.12670823
Five-gram model	LSTM model 4	0.02577319	0.03378378	0.12275664

Table 17: Best performing models of all data types for Task 2 by error column.

5 Discussion

There are two main aspects to be taken into consideration in this section. First, the results of the models on different dataset combinations are interpreted and commented on. After, the size and structure of the datasets are considered and analyzed.

5.1 Discussion of the results

From the results for the task of detecting error labels on the sentence level (Task 1), a first conclusion that can be drawn is that the models trained on Swedish BERT sentence-level embeddings yield significantly better results across all dataset types, regardless of whether they are fully original data or if they have been enhanced with non-authentic data (with the exception of the models making use of POS information, which were not trained using Swedish BERT embeddings). A possible reason for the better performance of models trained using pre-trained BERT embeddings could be the fact that Swedish BERT, like BERT in its other available languages, is trained on grammatically correct data. Therefore, it is likely that BERT captures enough grammatical knowledge that, when generating an embedding for a sentence containing grammatical errors, the error gets reflected and the model can detect and classify it with more ease.

Secondly, adding POS information to the automatically generated through the Embedding dimension word embeddings, counter-intuitively does not seem to provide relevant enough information for the models to yield better results. There is a possibility that the lower scores (relative to better performing models) were caused by the method of combining the tensors of the word embeddings with the tensors representing the POS tags, which was 'torch.add', an element-wise addition of tensors. Other ways of combining both representations for the model to learn from would need additional exploration.

Thirdly, continuing to consider the results of Task 1, especially the comparison of results from models trained on fully original learner data and those trained on original learner data with incorporated non-original incorrect sentences, it can be seen that the second type of models provide better results. This is the case of the model trained on original learner data with 500 incorporated FakeDaLAJ (S-FinV) sentences, which yields the highest F0.5 score overall (See Table 13) as well as across all error types (See Table 16). This raises the question of whether the inclusion of artificially generated error data into original learner data improves the performance of the models solely because of the increase in the general number of training sentences. By comparing the general results as well as the error type score distribution, one could say that an increase in the number of training sentences by adding non-original data does not necessarily yield better results, as the models trained on the dataset with 500 artificial sentences outputs higher F0.5 scores than those trained on the datasets with the additional 1,000 and 1,500 sentences respectively. This is true for all error labels except for the Syntactical one, which yields the best results with a model trained on the dataset including 1,500 extra sentences with Syntactical errors. However, that same model shows moderately lower scores than the model trained with the extra 500 synthetic sentences. One possibility is that, by adding 1,500 sentences containing exclusively errors of one type, and consequently increasing the representation of that error label, the trained model learns to detect that specific error type by neglecting its capacity to detect other types.

This leads the discussion to another question: should the different error types be equally represented in datasets for GED? From the results of the best model trained on 500 extra Syntactical errors one could deduct that, when the proportions of errors were of 2,100 correct sentences, 3,146 sentences with Lexical errors, 3,922 with Morphological, 2,688 with Orthographic, 1,843 with Punctuation and 4,263 with Syntactical, the best performing error types were mainly those with higher representation (that is, Syntactical and Morphological), and the lowest was the error type with the least amount of sentences (Punctuation). Moreover, the relation between number of sentences containing errors of a certain type and their F0.5 score

stands with the best model trained on the fully original dataset as well. The results from this master's thesis work seem to suggest that balancing the error label representation in datasets for GED could potentially improve performance on the prediction of grammatical errors, but confirming this hypothesis would require further research and the generation of synthetic error datasets for other error types. With these considerations, the third research question can be answered. With the positive results from combining only one synthetic error type into original data, it can be considered that exploration in the field of synthetic error data generation is promising. Considering the time limitations, this master's thesis is able to provide an insight into the capabilities of the usefulness of synthetic data. Keeping in mind that Swedish L2 data is at a position of inferiority compared to other languages, it seems like a field worth exploring.

A fourth discussion point is the considerably low F0.5 scores of the models trained to detect error labels on the column level (Task 2) as well as their high percentage of fully correct label predictions. As it can be seen in Table 14 for trigrams, Table 15 for five-grams and Table 17 for the score distribution by column, this experiment in the form that it has been designed might not be adequate to predict error types by column. The two highest performing models present high percentage of correct full label predictions, probably due to the fact that, when performing the task on the token level, the majority of tokens do not have an attached error label. Therefore, as mentioned by Leacock et al. (2014), a system that predicts all tokens as correct, when evaluated on a dataset where 10% of tokens are incorrect, will obtain an accuracy of 90%, and that does not signify that the model is well performing. This answers the second research question, on whether it is viable to work with error edit operations when performing GED. From the results in this work, one would say that, for this exercise to be more successful, it is likely that an intermediate task would be necessary, such as predicting the error labels by token (similarly to Task 1, but in more detail) disregarding the correction operation type. From a model able to perform that intermediate task, a new model could likely be developed to place the predicted errors into the three correction operation columns. This task is left for future exploration.

With the collection of points mentioned above, the first research question can be answered. The agreed ConLL-like format for the task poses a difficulty when intending to predict the correction operation tags (Task 2) due to the high frequency of tokens with no error tag. Tasks involving the detection of correction operations would probably benefit from being built up in different steps: binary detection (a token is either correct or incorrect), grammatical error classification (a token belongs to one or more of the five error labels), and finally edit operation classification (a token, to be correct, would need an addition, a deletion or a replacement).

5.2 Discussion of the dataset(s)

Regarding the resulting datasets from this master's thesis, the fully original learner data Swedish MuClaGED and its various expansions with synthetic data, two points need to be discussed. The first one is its small size, and the second one is the viability of the Swedish MuClaGED + FakeDaLAJ (S-FinV) hybrid combination.

It is apparent that the Swedish MuClaGED is of limited size (8,553 sentences coming from 502 student essays), especially considering that most tasks, not only in GED and NLP specifically but in the general field of machine learning, commonly require greater amounts of data to train models capable of producing satisfactory results. Other shared tasks using original learner data, such as the CoNLL-2013 shared task (Ng et al., 2013), provided annotated sentences coming from 1,414 essays, which is approximately three times the number of essays available in SweLL (Volodina et al., 2019), where the texts in MuClaGED come from. The BEA 2019 shared task (Bryant et al., 2019) offered their participants access to various corpora, containing 33,236 (from 3,600 texts), 1,037,561 and 57,151 sentences (from 1,400 essays) respectively, which are considerably larger than Swedish MuClaGED. However, as the goal of the Computational SLA shared task for which the MuClaGED dataset has been built is to offer the participants a dataset containing approximately 10,000 sentences of each participating language to construct a larger multilingual dataset, its

size is considered rather appropriate. Additionally, the Swedish MuClaGED has been proven to be extendable with 500 FakeDaLAJ (S-FinV) (Casademont Moner & Volodina, 2022) with a slight improvement in the results for error detection on the sentence level, namely Task 1 (See Section 3.3.3).

However, it must be taken into account that the combination of Swedish MuClaGED and FakeDaLAJ (S-FinV) implies the merging of two distinct error representation styles: the purely original format, where sentences with various errors are common, and the DaLAJ format (Volodina et al., 2021), which has modified sentences that contain only one error. This second style has the possibility of not being representative of the production of grammatical errors by language learners, which could lead to the production of models incapable of identifying real learner's errors, who are to benefit from Intelligent Computer-Assisted Language Learning (ICALL) systems. Although the results in this work show that an addition of 500 synthetic sentences of one error type to an original learner dataset improves the model's performance, the addition of non-original sentences of all grammatical error types into original data has not been yet tested.

6 Ethical considerations

Three considerations should be made when examining the ethical implications of this project: data privacy, possible bias and hypothetical use of the dataset.

In relation to the protection of the data privacy of the students who wrote the essays, it must be noted that the language in the SweLL dataset was pseudonymized to ensure the anonymity of participants (Megyesi et al., 2018). When producing the SweLL dataset, the authors carefully considered the European Union's General Data Protection Regulation (GDPR) (Regulation EU 2016/6791 from May 25, 2018) to avoid illegal and unethical use of data that is to be made publically accessible. Through pseudonymization, the authors process text that includes personal information in such a way that the personal data cannot be attributed to an individual unless additional information is used, and this additional information is kept separately, as specified by EU GDPR (Article 4). Therefore, the data used in this project, coming from SweLL, complies with the current EU privacy legislation. Additionally, taking into consideration that the MuClaGED dataset will be released for public use, extra precautions have been made such as shuffling the sentences to ensure the impossibility of reassembling full essays.

A second consideration is the possible bias in the dataset, both in terms of linguistic bias and in relation to the data representation across different language levels and the native tongue of the learners. Regarding linguistic bias, due to the nature of the data, which consists of essays where students write about themselves (e.g., their family, where they come from) or explain their opinion on a topic (e.g., opinion essays, reviews), it is understandable that linguistic bias will be present in the dataset to some extent. However, as the purpose of the dataset is to train models to perform grammatical error detection and, in principle, no natural language generation will be produced with this data, linguistic bias should not be an issue. On the other hand, the fact that the dataset is not balanced in terms of the representation of proficiency levels and native language of the students could possibly imply that the systems using models built from this dataset will be capable of detecting error patterns of the most represented learner groups more accurately than others. And this could have an effect on the experiences and learning outcomes of the students using these systems in the future.

The final consideration, related to the second one, is regarding the hypothetical use of this dataset for grading or evaluating students' work. It must be noted that the MuClaGED dataset for Swedish is limited to purely linguistic evaluation and does not check for semantics. For example, whether the student answered the question of the essay is not contemplated. However, since this project only aims to carry out the first experiments on the dataset, although there is awareness on this possible issue, there is no need to deal with it at the moment.

7 Conclusions

The principal product of this Master's thesis is the generation of the Swedish MuClaGED dataset, which has been designed specifically to fit in the field of Grammatical Error Detection (GED). The intention is for this dataset to be combined with datasets of other languages for a multilingual GED shared task that is expected to take place the spring of 2023. Swedish MuClaGED in its base form consists of a little over 8,500 sentences containing errors divided into a 5-type taxonomy across three edit operation columns. It allows for various error detection tasks involving the presence of grammatical errors on the sentence level and on the token level.

In this master's thesis, two possible tasks have been tested on Swedish MuClaGED: error label detection on the sentence level and error correction operation classification on the token level. The two experiments have yielded very different results in regards of the output models' capabilities. The first task has proven to be easier to generate satisfactory models, whereas the second task, although with high accuracy, has been found rather ineffective when detecting the position of errors in the columns. With the experiments, results for Swedish MuClaGED have been obtained to be provided to the participants in the shared task as tentative a baseline.

Additionally, an initial exploration on the possibility of producing a hybrid dataset of grammatical errors by combining original learner data with non-original synthetic data has been made. The tests which used MuClaGED with 500 sentences coming from FakeDaLAJ (S-FinV) have delivered promising results, which improve the scores of models built exclusively from Swedish MuClaGED on its own.

7.1 The research questions answered

As a summary, this subsection reintroduces the three initial research questions and answers them concisely.

The first question was: *Is a dataset with this agreed structure appropriate for the shared task it is aimed at? If there are any challenges, should the task be redefined?* The results on the task of error type detection on the sentence level show that this is a promising approach. However, the task of detecting correction operations directly on the token level poses a difficulty to train models due to the frequency of error-free tokens. A possible redesign of the task would be to perform it in incremental steps.

The second question is: *Is error edit operations a viable approach for working with Grammatical Error Detection?* It could potentially be a viable approach if performed at a different level than in Task 2 in this work. The task of identifying errors in terms of edit operations on the token level would likely benefit from simply performing the binary classification of errors, as well as from techniques other than n-grams.

And finally, the third question is: *Are the creation of synthetic error data and the production of hybrid datasets a research path worth taking in the future?* Having obtained positive results in the models for the task of error classification on the sentence level when using a hybrid original learner dataset with synthetic data, it is conclusive that exploring the field of synthetic error data generation is a promising option for the growth of Swedish L2 data.

7.2 Future work

As it has been mentioned in the discussion, it is hypothesized that an intermediate task previous to Task 2, error edit operation detection on the token level, would be necessary for this task to yield more appropriate results. Furthermore, the structure design of Swedish MuClaGED offers the possibility of more in-depth experiments which could be explored in the future.

In this work, only LSTM and Bi-LSTM models were trained for both tasks. However, considering the improvement in performance when working with BERT sentence-level embeddings, one would consider the employement of pre-trained models (either BERT itself or other transformer-based models) for the task. Similarly, other type of word embeddings could be explored within the same LSTM and Bi-LSTM structure, such as FastText, for example.

Moreover, the research on hybrid datasets of original texts written by students with artificially generated text is a field that has not been yet explored for Swedish L2 language. Extending FakeDaLAJ on different error types to enlarge MuClaGED to test its effectiveness when detecting grammatical errors seems to be a straightforward direction to be taken in the future, as the present results are suggestive of how hybrid datasets can boost GED models' performance.

References

Abiodun, O., Jantan, A., Omolara, O., Dada, K., Umar, A., Linus, O., Arshad, H., Aminu Kazaure, A., Gana, U., & Kiru, M. (2019). Comprehensive Review of Artificial Neural Network Applications to Pattern Recognition. *IEEE Access*, PP, 1–1.

Awasthi, A., Sarawagi, S., Goyal, R., Ghosh, S., & Piratla, V. (2019). Parallel Iterative Edit Models for Local Sequence Transduction. *CoRR*, abs/1910.02893.

Bell, S., Yannakoudakis, H., & Rei, M. (2019). Context is Key: Grammatical Error Detection with Contextual Word Representations. *CoRR*, abs/1906.06593.

Bryant, C., Felice, M., Andersen, Ø. E., & Briscoe, T. (2019). The BEA-2019 Shared Task on Grammatical Error Correction. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications* (pp. 52–75). Florence, Italy: Association for Computational Linguistics.

Bryant, C., Felice, M., & Briscoe, T. (2017). Automatic Annotation and Evaluation of Error Types for Grammatical Error Correction. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (pp. 793–805). Vancouver, Canada: Association for Computational Linguistics.

Casademont Moner, J. & Volodina, E. (Forthcoming, 2022). Generation of Synthetic Error Data of Verb Order Errors for Swedish. *17th Workshop on Innovative Use of NLP for Building Educational Applications*.

Dahlmeier, D., Ng, H. T., & Wu, S. M. (2013). Building a Large Annotated Corpus of Learner English: The NUS Corpus of Learner English. In *Proceedings of the Eighth Workshop on Innovative Use of NLP for Building Educational Applications* (pp. 22–31). Atlanta, Georgia: Association for Computational Linguistics.

Dale, R., Anisimoff, I., & Narroway, G. (2012). HOO 2012: A Report on the Preposition and Determiner Error Correction Shared Task. In *Proceedings of the Seventh Workshop on Innovative Use of NLP for Building Educational Applications* (pp. 54–62).: Association for Computational Linguistics.

Dale, R. & Kilgarriff, A. (2011). Helping Our Own: The HOO 2011 Pilot Shared Task. In *Proceedings of the 13th European Workshop on Natural Language Generation* (pp. 242–249). Nancy, France: Association for Computational Linguistics.

Devlin, J., Chang, M., Lee, K., & Toutanova, K. (2018). BERT: Pre-training of Deep Bidirectional Transformers for Language ´Understanding. *CoRR*, abs/1810.04805.

Evert, S. (2004). The Statistics of Word Cooccurrences: Word Pairs and Collocations.

Grundkiewicz, R., Junczys-Dowmunt, M., & Heafield, K. (2019). Neural Grammatical Error Correction Systems with Unsupervised Pre-training on Synthetic Data. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications* (pp. 252–263). Florence, Italy: Association for Computational Linguistics.

Hajič, J., Ciaramita, M., Johansson, R., Kawahara, D., Martí, M. A., Màrquez, L., Meyers, A., Nivre, J., Padó, S., Štěpánek, J., Straňák, P., Surdeanu, M., Xue, N., & Zhang, Y. (2009). The CoNLL-2009 Shared Task: Syntactic and Semantic Dependencies in Multiple Languages. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL 2009): Shared Task* (pp. 1–18). Boulder, Colorado: Association for Computational Linguistics.

Hochreiter, S. & Schmidhuber, J. (1997). Long Short-Term Memory. *Neural computation*, 9, 1735–80.

Junczys-Dowmunt, M. & Grundkiewicz, R. (2016). Phrase-based Machine Translation is State-of-the-Art for Automatic Grammatical Error Correction. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing* (pp. 1546–1556). Austin, Texas: Association for Computational Linguistics.

Kasewa, S., Stenetorp, P., & Riedel, S. (2018). Wronging a Right: Generating Better Errors to Improve Grammatical Error Detection. *CoRR*, abs/1810.00668.

Leacock, C., Chodorow, M., Gamon, M., & Tetreault, J. (2014). *Automated Grammatical Error Detection for Language Learners, Second Edition*, volume 7.

Liu, Z. & Liu, Y. (2017). Exploiting Unlabeled Data for Neural Grammatical Error Detection. *Journal of Computer Science and Technology*, 32(4).

Madi, N. & Al-Khalifa, H. S. (2018a). Grammatical Error Checking Systems: A Review of Approaches and Emerging Directions. *2018 Thirteenth International Conference on Digital Information Management (ICDIM)*, (pp. 142–147).

Madi, N. & Al-Khalifa, H. S. (2018b). A Proposed Arabic Grammatical Error Detection Tool Based on Deep Learning. *Procedia computer Science, volume 142, p. 352-355*.

Megyesi, B., Granstedt, L., Johansson, S., Prentice, J., Rosén, D., Schenström, C.-J., Sundberg, G., Wirén, M., & Volodina, E. (2018). Learner Corpus Anonymization in the Age of GDPR: Insights from the Creation of a Learner Corpus of Swedish. In *Proceedings of the 7th workshop on NLP for Computer Assisted Language Learning* (pp. 47–56). Stockholm, Sweden: LiU Electronic Press.

Moore, R. C. (2002). Fast and Accurate Sentence Alignment of Bilingual Corpora. In *Proceedings of the 5th Conference of the Association for Machine Translation in the Americas: Technical Papers* (pp. 135–144). Tiburon, USA: Springer.

Ng, H. T., Wu, S. M., Briscoe, T., Hadiwinoto, C., Susanto, R. H., & Bryant, C. (2014). The CoNLL-2014 Shared Task on Grammatical Error Correction. In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task* (pp. 1–14). Baltimore, Maryland: Association for Computational Linguistics.

Ng, H. T., Wu, S. M., Wu, Y., Hadiwinoto, C., & Tetreault, J. (2013). The CoNLL-2013 Shared Task on Grammatical Error Correction. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning: Shared Task* (pp. 1–12). Sofia, Bulgaria: Association for Computational Linguistics.

Nivre, J., Megyesi, B., Gustafson-Capková, S., Salomonsson, F., & Dahlqvist, B. (2008). Cultivating a Swedish treebank. *Resourceful language technology: Festschrift in honor of Anna Sågvall Hein*, (pp. 111–120).

Rei, M. & Yannakoudakis, H. (2016). Compositional Sequence Labeling Models for Error Detection in Learner Writing. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (pp. 1181–1191). Berlin, Germany: Association for Computational Linguistics.

Rekathati, F. (2021). The KBLab Blog: Introducing a Swedish Sentence Transformer.

Rudebeck, L. & Sundberg, G. (2021). *SweLL correction annotation guidelines.* Technical report, GU-ISS Research report series, Department of Swedish, University of Gothenburg. http://hdl.handle.net/2077/69434.

Sarker, I. (2021). Deep Learning: A Comprehensive Overview on Techniques, Taxonomy, Applications and Research Directions.

Stahlberg, F. & Kumar, S. (2021). Synthetic Data Generation for Grammatical Error Correction with Tagged Corruption Models. *CoRR*, abs/2105.13318.

Straka, M. (2018). UDPipe 2.0 Prototype at CoNLL 2018 UD Shared Task. In *CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies* (pp. 197–207).

Volodina, E., Granstedt, L., Matsson, A., Megyesi, B., Pilán, I., Prentice, J., Rosén, D., Rudebeck, L., Schenström, C.-J., Sundberg, G., & Wirén, M. (2019). The SweLL Language Learner Corpus: From Design to Annotation. *Northern European Journal of Language Technology*.

Volodina, E., Mohammed, Y. A., & Klezl, J. (2021). DaLAJ - A Dataset for Linguistic Acceptability Judgments for Swedish: Format, baseline, sharing. *Proceedings of the 10th Workshop on Natural Language Processing for Computer Assisted Language Learning (NLP4CALL 2021). Linköping Electronic Conference Proceedings 177:3, s. 28-37*.

Volodina, E., Pilán, I., Eide, S. R., & Heidarsson, H. (2014). You get what you annotate: a pedagogically annotated corpus of coursebooks for Swedish as a Second Language. *Proceedings of the third workshop on NLP for computer-assisted language learning*.

Xue, H. & Hwa, R. (2014). Improved ´Correction Detection in Revised ESL Sentences. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)* (pp. 599–604). Baltimore, Maryland: Association for Computational Linguistics.

Yannakoudakis, H., Briscoe, T., & Medlock, B. (2011). A New Dataset and Method for Automatically Grading ESOL Texts. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies* (pp. 180–189). Portland, Oregon, USA: Association for Computational Linguistics.

Yannakoudakis, H., Rei, M., Andersen, Ø. E., & Yuan, Z. (2017). Neural Sequence-Labelling Models for Grammatical Error Correction. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing* (pp. 2795–2806). Copenhagen, Denmark: Association for Computational Linguistics.

Yasunaga, M., Leskovec, J., & Liang, P. (2021). LM-Critic: Language Models for Unsupervised Grammatical Error Correction. *CoRR*, abs/2109.06822.

Yuan, Z., Taslimipoor, S., Davis, C., & Bryant, C. (2021). Multi-Class Grammatical Error Detection for Correction: A Tale of Two Systems. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing* (pp. 8722–8736). Online and Punta Cana, Dominican Republic: Association for Computational Linguistics.

Zhao, Y., Jiang, N., Sun, W., & Wan, X. (2018). Overview of the NLPCC 2018 Shared Task: Grammatical Error Correction. In *NLPCC*.

8 Appendices

8.1 Scores of the best performing model on original MuClaGED

	Bi-LSTM model 5 with BERT sentence-level embeddings on original MuClaGED					
	Correct full label 25.73%					
	Overall results	**Lexical**	**Morphological**	**Orthographic**	**Punctuation**	**Syntactical**
Accuracy	0.713216	0.679532	0.661988	0.743859	0.796491	0.684210
F1 score	0.589692	0.547854	0.630906	0.561122	0.412162	0.664179
Precision	0.586551	0.549668	0.589498	0.585774	0.5	0.695312
Recall	0.592866	0.546052	0.678571	0.678571	0.350574	0.635714
F0.5 score	0.587803	0.548941	0.605392	0.605392	0.460725	0.646802

8.2 Scores of the best performing model on MuClaGED + 500 FakeDaLAJ (S-FinV)

	Bi-LSTM model 5 with BERT sentence-level embeddings o MuClaGED + 500 FakeDaLAJ					
	Correct full label 30.82%					
	Overall results	**Lexical**	**Morphological**	**Orthographic**	**Punctuation**	**Syntactical**
Accuracy	0.761419	0.733924	0.698447	0.792682	0.847006	0.735033
F1 score	0.654906	0.619047	0.682983	0.607966	0.485074	0.729943
Precision	0.629081	0.598159	0.611691	0.611814	0.625	0.677148
Recall	0.682943	0.641447	0.773087	0.604166	0.396341	0.791666
F0.5 score	0.639163	0.606343	0.638344	0.610269	0.560344	0.697322