Swedish MuClaGED: A new dataset for Grammatical Error Detection in Swedish

Judit Casademont Moner
University of Gothenburg / Sweden
guscasaju@student.gu.se

Elena Volodina
University of Gothenburg / Sweden
elena.volodina@svenska.gu.se

Abstract

This paper introduces the Swedish MuClaGED[1] dataset, a new dataset specifically built for the task of Multi-Class Grammatical Error Detection (GED). The dataset has been produced as a part of the *multilingual Computational SLA*[2] shared task initiative[3]. In this paper we elaborate on the generation process and the design choices made to obtain Swedish MuClaGED. We also show initial baseline results for the performance on the dataset in a task of Grammatical Error Detection and Classification on the sentence level, which have been obtained through (Bi)LSTM ((Bidirectional) Long-Short Term Memory) methods.

1 Introduction

Due to high migration of people around the globe, learning a language of a new country of residence is increasingly important, and educational applications such as grammar checkers and other evaluation tools suitable for language learners are increasingly in demand. Grammatical Error Correction (GEC) (e.g. Omelianchuk et al., 2020; Bryant et al., 2019) and Grammatical Error Detection (GED) (e.g. Yuan et al., 2021; Daudaravicius et al., 2016) are two well-established fields in NLP that focus on techniques to support development of language users' writing skills, where errors are flagged (detection) and suggestions for corrections are generated (correction) - often in synchronous mode, i.e. as the user writes (e.g. Ranalli and Yamashita, 2022).[4] However, correction of errors without explaining reasons behind corrections does not necessarily lead to effective learning, and we argue therefore that GED **and** subsequent classification of errors by an error type constitute a critical first step for generation of meaningful corrective feedback.

Within Second Language Acquisition (SLA), *corrective feedback* can be defined as the teacher's identification of an error and subsequent attempt(s) to inform the learner about it in some way (Chaudron, 1988). The research in the field has moved from an earlier position where corrective feedback was considered unhelpful for language learning (Krashen, 1981) to the current understanding that corrective feedback can indeed be important and sometimes even crucial for adult learners to advance in the second/foreign language (Van Beuningen et al., 2012; Lyster et al., 2013). Research on the topic is based on the firm assumption that corrective feedback is necessary for second language learners. And recent studies have focused on the quality of automatic error detection and classification, as well as the best ways of providing feedback - among others, on the timing of said feedback (i.e. synchronous versus asynchronous) and on its effects on the cognitive process, e.g. Ranalli and Yamashita (2022).

In view of that, the Computational SLA team has considered **error detection and classification**, as the main focus for a shared task, which we argue should be given more attention.

1.1 MuClaGED task in a nutshell

The task has been defined as a *multi-lingual multi-class grammatical error detection in low-resource contexts*. One of the important principles is that the data should be *authentic language learner data*. Many current grammar checkers have been trained on texts produced by native speakers (L1) or on the language produced by advanced non-native speakers in highly academic texts, such as in the case of the Helping Our Own (HOO) shared

[1] MuClaGED = Multi-Class Grammatical Error Detection
[2] SLA = Second Language Acquisition
[3] Note that this version of the dataset is preliminary. The final guidelines for the dataset and the task may change as a result of the current experiments and further work on the definition of the task and datasets. However, the current dataset will be made available as such for the community one the shared task is over.
[4] Interrelation of the two fields is well-captured by Google Ngrams, even though we realize that the corpus is decisive for this type of generalizations: https://tinyurl.com/bddadpus

Judit Casademont Moner and Elena Volodina. Swedish MuClaGED: A new dataset for Grammatical Error Detection in Swedish. *Proceedings of the 11th Workshop on Natural Language Processing for Computer Assisted Language Learning (NLP4CALL 2022)*. Linköping Electronic Conference Proceedings 190: 36–45.

task (Dale and Kilgarriff, 2011). Intuitively, these systems are not as well suited for Intelligent Computer-Assisted Language Learning (ICALL) or Automatic Writing Evaluation (AWE) systems. Indeed, Leacock et al. (2014) have convincingly shown that foreign language learners' error correction and feedback will benefit from models trained on real L2 students' texts. Hence the importance of using *authentic language learner data*.

Another principle is the *focus on low-represented languages*. Both GEC and GED have been mainly researched on the basis of English data. Therefore, shared tasks on other, less-represented languages are needed to stimulate further research. However, unlike English, many other languages have smaller datasets of error-annotated L2 data compared to English. Therefore, the Computational SLA team has initiated a multi-lingual task where several language datasets, potentially small ones, should be unified in format and annotation for the shared task with a possibility to augment data, and/or use datasets from several languages through domain adaptation, transfer learning, and other modern techniques. Swedish, as one of the less-represented languages, is a part of this task, alongside Czech, Italian, German and English.

The teams will have sentence-scrambled authentic learner data with the task to develop methods for the following:

(1) binary classification on a sentence level (correct – incorrect)

(2) binary classification on a token level (correct – incorrect)

(3) error classification on a sentence level (5-class taxonomy)

(4) error classification on a token level (5-class taxonomy)

The results will be evaluated using recall, precision, accuracy and F-scores per the target language, and teams will have a possibility to use additional data in addition to the one provided by the organizers.

In other words, 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 multi-class error detection, that is, classifying detected errors into five main categories (Punctuation, Orthographic, Lexical, Morphological and Syntactic). The five categories have been defined broadly, so that all languages could convert their tagsets to produce comparable annotations.

The task is aimed at promoting a few languages which have not been in much focus for GED or GEC, and where appropriately annotated datasets are available, even if modest in size. Therefore the size of the datasets is limited to 10,000 sentences, imitating the low-resource context even where more data is available. The latter does not, however, apply to Swedish since error-annotated Swedish data contains only approximately 8,500 sentences from learner essays (including correct ones).

This will be the first time that original L2 learner data for Swedish will be used in a shared task focusing on GED. The main focus of this article is to present the generation process of the Swedish GED dataset necessary for the MuClaGED shared task according to specifications agreed on between the task organizers. In Section 3 we describe the resulting dataset. Additionally, we present an initial experiment on the resulting dataset to explore and evaluate its functionality in the task of Grammatical Error Detection and Classification on the sentence level (task (3) above) and to present the first baseline for the task (Section 4).

2 Related work

2.1 Grammatical Error Detection and Classification

Grammatical Error Detection (GED) is a challenging task in NLP which has gained considerable attention in the recent years. It is generally 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 and Al-Khalifa, 2018a), as well as to assess their work (Rei and Yannakoudakis, 2016). The need for these tools exists in all languages, even in languages with a notable 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 and 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).

2.2 Approaches to GED and GEC

Over time, various attempts have been made to address the task of detecting grammatical errors in written text. As presented by Madi and Al-Khalifa (2018a), 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).

One popular approach to tackle the task is using deep learning models, such as Neural Networks (NN) or Recurrent Neural Networks (RNN), 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 and Al-Khalifa, 2018b).

With the recent arrival of transformers in the field of NLP, 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.

2.3 Data required for GED

Obtaining useful data for the tasks of GED and GEC can be challenging, especially when the desired approaches are statistical or machine learning methods, which require large quantities of labeled data. Written error data to perform GED and GEC 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).

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 intended for L2 students. 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 and for general use are usually extracted from formal and correct sources such as news sites or encyclopedias. The written texts' language style found in these corpora is usually different from the one used by students in their essays or other language learning tasks. This means that it is possible that a language model trained on encyclopedic text will not perform accurately GED on L2 students' texts. Therefore, we consider the use of original learner data the right choice for the task at hand.

3 Constructing MuClaGED

3.1 Original learner data

The source of the data used to craft the Swedish MuClaGED dataset is the SweLL (Swedish

Learner Language) gold corpus (Volodina et al., 2019). SweLL is an infrastructure with several linguistic datasets, 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, a research unit and a part of the Nationella Språkbanken (the National Language Bank of Sweden), with the purpose to set the foundations of the research on Second Language Acquisition (SLA) on the Swedish language (Rudebeck and 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 includes transcription (from hand-written originals), pseudonymization (to ensure the writers' privacy) and normalization (i.e. rewriting to correct version), 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"[5] 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, and only a few are kept in MuClaGED.

The SweLL-gold correction taxonomy consists of 29 error correction tags, which can be grouped into the following six subgroups: Punctuation, Orthographic, Lexical, Morphological, Syntactical and Other. For this work, we consider the first five categories, namely POLMS. Furthermore, the Other category represents comments and tags 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 guidelines (Rudebeck and Sundberg, 2021).

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

3.2 From SweLL-gold to MuClaGED

In this project, we transform the existing original learner data in Swedish, the SweLL-gold dataset (Volodina et al., 2019), into the CoNLL-like format agreed on by the Computational SLA team to build Swedish MuClaGED, exemplified in Fig. 2.

The format specifications go as follows. The established taxonomy contains five error categories, to be distributed into three correction operation types, represented in columns. These error categories are the top error tags used in Volodina et al. (2019), namely POLMS (**P**unctuation, **O**rthographic, **L**exical, **M**orphologic and **S**yntactic). 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.

In practical terms it means that, for example, if a sentence contains a misspelled word, the edit operations would be 'R' – replacement, and the error type be 'O' – orthographic. The 'O' code will be filled into the column 'R' for that particular token. If the same word is involved in some other error types, e.g. morphological agreement, a code 'M' – morphological error – will be added into the same column 'R'.

Additions are attached to the token after, where the additional necessary token (or tokens) should be added to render the sentence grammatically correct. For this purpose, a dummy token ('@') is added as the last position of the sentence, to store 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

(1) four comment-lines containing (i) the original sentence ('text'), (ii) the corrected version of the sentence ('corrected_text'), (iii) sentence id ('sent_id') and (iv) the metadata with level, first language ('L1') and the data split (80% is'Train', 10% is 'Dev', and 10% is 'Test'); during the development period we kept essay ids for potential need to double-check with the full essays. For the shared task essay ids are unnecessary and will be removed.

(2) one line per token with the token index, the word itself and the three edit operation columns (Addition, Deletion, Replacement). The columns are filled with corresponding error type(s) that have undergone that particular editing operation.

To complete the format transformation of the

```
# text = Vi fortfarande behöver vårt bibliotek ! @
# corrected_text = Vi behöver fortfarande vårt bibliotek ! @
# sent id = 86
# metadata = Approximate level = Nybörjare, L1 = Berberspråk, Marockansk arabiska, Split = Train
1       Vi          _       _       _
2       fortfarande     _       _       ['S']
3       behöver         _       _       _
4       vårt        _       _       _
5       bibliotek       _       _       _
6       !       _       _       _
7       @       _       _       _
```

Figure 1: Sentence example of the dataset with a word order error. The columns, in order, are: token id, token, addition errors, deletion errors, replacement errors. Translation: 'We still need our library!'

```
# text = Jag kunde inte forbi med de så var ensama . @
# corrected_text = Jag kunde inte lämna dem så de var ensamma . @
# sent id = 114
# metadata = 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 couldn't leave them so they were alone.'

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 (Part of Speech) information and gathering metadata.

The first step to obtain Swedish MuClaGED from SweLL-gold is to reach a sentence level alignment between the original text ('source') and its corrected version ('target'). The goal of this step is to distribute the error labels into the proper error correction operation type, represented in the ADR columns, specifically the additions and the deletions. When aligning the original text and the corrected text, we obtain 3 possible situations: one-to-one matches, where one token in the source corresponds to one token in the target; one-to-zero (no matches), where a token or more in the source cannot be matched to the target, or vice-versa, zero-to-one; and matches on different number of tokens (many-to-one, one-to-many and many-to-many). 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.

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, or by 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 error label distribution, also referred to as 'label logic' is shown in Table 1.

Finally, the error tags are simplified so that we are left with only the five main categories (Lexical, Morphological, Orthographic, Punctuation and Syntactic) by removing the second part of the original tags. The 35 labels found in the SweLL

Operation types	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 1: Error label distribution by error operation column.

taxonomy (Volodina et al., 2019) are thus reduced to five categories according to the first head category of the SweLL tag (e.g., 'M-Verb', which represents Morphological errors involving verbs and auxiliaries, becomes simply 'M'). Error corrections spanning a group of tokens receive numbering starting from the second token. This is done by adding ':2' (and consecutively) to the error tag.

To make sure that the dataset can be shared with any team willing to participate in the shared task despite the GDPR restrictions, (1) metadata was restricted to two labels - levels of the course and mother tongues; and (2) essays were scrambled and sentences were ordered randomly to limit a possibility to reconstruct original essays.

3.3 The resulting dataset

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 representas as follows: an initial comment-line with the full text, a second comment-line with the corrected text, a third comment-line with the sentence id and a fourth comment-line with metadata, containing the approximate level of the student, their native language or languages and the split the sentence belongs to (either train, test or dev splits, which represent 80%, 10% and 10% of the dataset respectively). The comment-lines are followed by one line for each individual token in the sentence. The token-level information consists of three error correction operation categories, namely ADR, standing for **A**ddition, **D**eletion and **R**eplacement.

In the dataset we can find the following error distribution by token (Table 2). One sentence might contain more than one error of the same type, and one token might be involved in more than one type of errors 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 2: Error distribution by token

Table 3 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 3: Error distribution by sentence

Table 4 shows the error distribution by correction operation categories (Addition, Deletion or Replacement).

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

Table 4: Error counts by column

4 Baseline experiments on MuClaGED

4.1 Methods

The experiment was carried out by using LSTMs and Bi-LSTMs on simple word embeddings, word embeddings combined with POS tags information and on Swedish BERT sentence embeddings (Rekathati, 2021).

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

```
# text = Tre barn har jag Jag tyckem min lägenhet . @
# corrected_text = Tre barn har jag. Jag tycker om min lägenhet . @
# sent id = 103
# metadata = 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        ['S']   _       _
9       .       _       _       _
10      @       _       _       _
```

Figure 3: 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.'

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.

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 experimentation purposes, both LSTMs and Bi-LSTMs were employed in this work.

The conscious choice was made to use LSTMs and Bi-LSTMs for this project instead of a transformer-based approach, which currently tends to yield the most promising results. We consider the main focus of this work to be the generation of a new dataset specifically designed for the task of grammatical error classification and not the obtainment of state-of-the-art results, as that will be the goal for the participants in the eventual shared task.

Swedish BERT sentence embeddings Bidirectional Encoder Representations from Transformers, commonly referred to 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 by learning 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, including Swedish.

For this work, we decide 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.

4.2 MuClaGED classification experiment

A machine learning experiment was performed on the generated Swedish MuClaGED dataset, with the goal (i) to test the functionality of the generated dataset for a possible task in the field of GED, and (ii) to obtain tentative baseline results for the planned shared task, to compare the participants'

scores against.

This 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'. Therefore, we are not using the token-level information for this evaluation task, we are exclusively testing the capacity of the model to detect the presence of errors and to classify them into five categories.

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.

4.3 Results of the experiments

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. However, for these initial baseline results, to decide on the best performing model, we consider mainly the F0.5-score, although other metrics such as accuracy, precision and recall will also be considered as evaluation metrics in the shared task.

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 $\geq$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).

Table 5 shows the two best performing models for the two data representations used for the experiments: the original L2 data and the original L2 data with POS information added. Only the results for the Bi-LSTM models are shown because, in both cases, the results improved notably when using a Bi-LSTM compared to using an LSTM.

	L2 data BERT Bi-LSTM	**L2 data with POS BERT Bi-LSTM**
Lexical	0.54894179	0.44901065
Morphological	0.60539215	0.51724137
Orthographic	0.57565789	0.33475783
Punctuation	0.46072507	0.05263157
Syntactic	0.64680232	0.55408472

Table 5: Best performing models by error type.

5 Concluding remarks

We have presented Swedish MuClaGED, one of five language datasets for the shared task on multiclass error detection. The dataset was evaluated for the task and we have reported the baseline results.

The main limitation is the size of the dataset. 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. Therefore, there is the possibility that, for certain purposes, especially if the dataset is to be used on its own, the quantity of data present, of 8,553 sentences, might not suffice. 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, we consider its size to be rather appropriate.

A second 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). Although, overall, the results in Table 5 show no remarkably large difference in performance, we consider it relevant to note that the error types that show the highest f0.5-scores correspond with the error types with more representation in the dataset, and vice versa.

Regarding the results of the experimentation on the task of Grammatical Error Detection and Classification on the sentence level, a first conclusion that can be drawn is that the models trained on Swedish BERT sentence-level embeddings yield significantly better general results. 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.

The results on the task of error type detection on the sentence level show that the proposed format of the dataset and the approach chosen for the experiment are promising. Additionally, 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.

Experiments with synthetic error datasets to complement MuClaGED have been initiated and shown very promising results (Casademont Moner, 2022). Synthetic data could have helped reach the necessary level of 10,000 sentences for the shared task and reduce the imbalance of underrepresented error types. However, this work is outside the scope of this article and the type of the data that the shared task requires.

Finally, the requirement on cross-language similarity/comparability of the datasets in the shared task (with regards to labels) might require additional changes and modifications to the presented dataset before the final version is adopted. The presented experiment and dataset are to be viewed as a proof-of-concept.

Acknowledgments

This work has been supported by a research grant from the Swedish Riksbankens Jubileumsfond *SweLL - research infrastructure for Swedish as a second language*, IN16-0464:1; by *Nationella språkbanken*, funded by the Swedish Research Council (2018-2024, contract 2017-00626) and their participating partner institutions; and by VINNOVA through its funding of *SwedishGlue: a benchmark suite for language models* (2020-02523).

References

Oludare Abiodun, Aman Jantan, Oludare Omolara, Kemi Dada, Abubakar Umar, Okafor Linus, Humaira Arshad, Abdullahi Aminu Kazaure, Usman Gana, and Muhammad Kiru. 2019. Comprehensive Review of Artificial Neural Network Applications to Pattern Recognition. *IEEE Access*, PP:1–1.

Samuel Bell, Helen Yannakoudakis, and Marek Rei. 2019. Context is Key: Grammatical Error Detection with Contextual Word Representations. *CoRR*, abs/1906.06593.

Christopher Bryant, Mariano Felice, Øistein E. Andersen, and Ted Briscoe. 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, pages 52–75, Florence, Italy. Association for Computational Linguistics.

Christopher Bryant, Mariano Felice, and Ted Briscoe. 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)*, pages 793–805, Vancouver, Canada. Association for Computational Linguistics.

Judit Casademont Moner. 2022. Multi-Class Grammatical Error Detection: Data, Benchmarks and Evaluation Metrics for the First Shared Task on Swedish L2 Data. Master's thesis, University of Gothenburg.

Craig Chaudron. 1988. *Second Language Classrooms. Research on Teaching and Learning.* Cambridge University Press, New York.

Robert Dale and Adam Kilgarriff. 2011. Helping Our Own: The HOO 2011 Pilot Shared Task. In *Proceedings of the 13th European Workshop on Natural Language Generation*, pages 242–249, Nancy, France. Association for Computational Linguistics.

Vidas Daudaravicius, Rafael E. Banchs, Elena Volodina, and Courtney Napoles. 2016. A report on the automatic evaluation of scientific writing shared task. In *Proceedings of the 11th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 53–62, San Diego, CA. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language ´Understanding. *CoRR*, abs/1810.04805.

Stephen Krashen. 1981. Second language acquisition. *Second Language Learning*, 3(7):19–39.

Claudia Leacock, Martin Chodorow, Michael Gamon, and Joel Tetreault. 2014. *Automated Grammatical Error Detection for Language Learners, Second Edition*, volume 7.

Roy Lyster, Kazuya Saito, and Masatoshi Sato. 2013. Oral corrective feedback in second language classrooms. *Language teaching*, 46(1):1–40.

Nora Madi and Hend S. Al-Khalifa. 2018a. A Proposed Arabic Grammatical Error Detection Tool Based on Deep Learning. *Procedia computer Science, volume 142, p. 352-355*.

Nora Madi and Hend Suliman Al-Khalifa. 2018b. Grammatical Error Checking Systems: A Review of Approaches and Emerging Directions. *2018 Thirteenth International Conference on Digital Information Management (ICDIM)*, pages 142–147.

Kostiantyn Omelianchuk, Vitaliy Atrasevych, Artem Chernodub, and Oleksandr Skurzhanskyi. 2020. Gector–grammatical error correction: tag, not rewrite. *arXiv preprint arXiv:2005.12592*.

Jim Ranalli and Taichi Yamashita. 2022. Automated written corrective feedback: Error-correction performance and timing of delivery. *Language Learning & Technology*, 26(1):1–25.

Marek Rei and Helen Yannakoudakis. 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)*, pages 1181–1191, Berlin, Germany. Association for Computational Linguistics.

Faton Rekathati. 2021. The KBLab Blog: Introducing a Swedish Sentence Transformer.

Lisa Rudebeck and Gunlög Sundberg. 2021. SweLL correction annotation guidelines. Technical report, GU-ISS Research report series, Department of Swedish, University of Gothenburg. http://hdl.handle.net/2077/69434.

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

Catherine G Van Beuningen, Nivja H De Jong, and Folkert Kuiken. 2012. Evidence on the effectiveness of comprehensive error correction in second language writing. *Language learning*, 62(1):1–41.

Elena Volodina, Lena Granstedt, Arild Matsson, Beáta Megyesi, Ildikó Pilán, Julia Prentice, Dan Rosén, Lisa Rudebeck, Carl-Johan Schenström, Gunlög Sundberg, and Mats Wirén. 2019. The SweLL Language Learner Corpus: From Design to Annotation. *Northern European Journal of Language Technology*.

Helen Yannakoudakis, Ted Briscoe, and Ben Medlock. 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*, pages 180–189, Portland, Oregon, USA. Association for Computational Linguistics.

Zheng Yuan, Shiva Taslimipoor, Christopher Davis, and Christopher Bryant. 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*, pages 8722–8736, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.