Exploring Linguistic Acceptability in Swedish Learners' Language

Julia Klezl, Yousuf Ali Mohammed, Elena Volodina
University of Gothenburg, Sweden
name.surname1.surname2@svenska.gu.se

Abstract

We present our initial experiments on binary classification of sentences into linguistically correct versus incorrect ones in Swedish using the DaLAJ dataset (Volodina et al., 2021a). The nature of the task is bordering on linguistic acceptability judgments, on the one hand, and on grammatical error detection task, on the other. The experiments include models trained with different input features and on different variations of the training, validation, and test splits. We also analyze the results focusing on different error types and errors made on different proficiency levels. Apart from insights into which features and approaches work well for this task, we present first benchmark results on this dataset. The implementation is based on a bidirectional LSTM network and pretrained FastText embeddings, BERT embeddings, own word and character embeddings, as well as part-of-speech tags and dependency labels as input features. The best model used BERT embeddings and a training and validation set enriched with additional correct sentences. It reached an accuracy of 73% on one of three test sets used in the evaluation. These promising results illustrate that the data and format of DaLAJ make a valuable new resource for research in acceptability judgements in Swedish.

1 Introduction

Linguistic acceptability comes from the field of generative linguistics. It is based on native speakers' intuitive judgements of whether a sentence is acceptable or not (Schütze, 1996). While Lau et al. (2017) argue that acceptability is a gradient phenomenon, it generally is treated as a binary classification task (Warstadt et al., 2019). To create datasets for acceptability judgements, either existing incorrect sentences are collected, for example from linguistic literature (Lau et al., 2017; Lawrence et al., 2000), or correct sentences are manipulated (Marvin and Linzen, 2018). Using incorrect sentences by language learners has not been a common approach in this field so far.

There have been several studies on linguistic acceptability in English over the last years, using various forms of neural networks, targeting different error types, and focusing on different underlying aims. Neural networks trained to make acceptability judgements can yield for example theoretical insights into how language is perceived and acquired (Lawrence et al., 2000; Lau et al., 2017), or into what knowledge language models represent (Linzen et al., 2016; Jing et al., 2019). Practical applications of such models include evaluation of results from language-generating systems (such as question-answering or machine translation) or providing assistance in language learning.

Contrary to the field in English, we are aware of only one study on linguistic acceptability on the Swedish language (Taktasheva et al., 2021), where authors use synthetically manipulated data focusing on effects of word order errors on model predictions. Our study is inspired by the research on linguistic acceptability, however, we set it into the domain of second language acquisition. We formulate the task as a binary classification on a sentence level, similar to Daudaravicius et al. (2016), where the system output should classify a sentence as correct or incorrect (i.e. containing an error). We see this type of classification as a first step to future grammatical error detection (GED) and correction (GEC) systems for Swedish, and as a first step before generating feedback on errors.

In our work, we present an exploration of the binary sentence classification task on DaLAJ, a Dataset for Linguistic Acceptability in Swedish, where each sentence pair contains (1) a sentence with one error only and (2) a corrected sentence. Due to the fact that the dataset is new, and the task unprecedented in this form for Swedish, our study has a strong exploratory character. Our contributions include a first evaluation of the strengths,

Julia Klezl, Yousuf Ali Mohammed and Elena Volodina. Exploring Linguistic Acceptability in Swedish Learners' Language. *Proceedings of the 11th Workshop on Natural Language Processing for Computer Assisted Language Learning (NLP4CALL 2022)*. Linköping Electronic Conference Proceedings 190: 84–94.

possibilities, and certain drawbacks of the dataset, a comparison of different input features to the neural network, and first benchmark results for this task.

In the next section, we briefly outline two comparable studies in English. In section 3, the data, features, and models are introduced, followed by the results in section 4, as well as a discussion and a conclusion with some ideas for future work in sections 5 and 6.

2 Related work

Comparing acceptability models is generally difficult, since there are big differences across languages, target errors, metrics and datasets. The following shared task and study are relatively similar in set-up and aim to our focus, so they provide some context to view our work in.

2.1 AESW 2016

The goal in the Automatic Evaluation of Scientific Writing shared task (AESW) 2016 was to identify sentences in need of correction in scientific articles written in English (Daudaravicius et al., 2016). This did not only include grammatical errors but also stylistic features inappropriate for the academic genre. Predictions were given both in a binary and a probabilistic version. The task organizers report that six teams participated, two of which used deep learning methods, two maximum entropy, and the remaining two logistic regression and support vector machines. The teams using deep learning ranked highest with F1-scores of 61.08 and 62.78 on the binary task (Daudaravicius et al., 2016). One of them used a convolutional neural network and pretrained word embeddings (Lee et al., 2016). The other team combined several character - and one word-based encoder-decoder models and a sentence-level convolutional layer by majority vote (Schmaltz et al., 2016).

2.2 CoLA

CoLA is the **Co**rpus of **L**inguistic **A**cceptability, a collection of ”10,657 English sentences labeled as grammatical or ungrammatical from published linguistics literature” (Warstadt et al., 2019). It targets morphological, syntactic, and semantic errors. The authors also present first models trained on this dataset. The most successful one uses transfer learning with an encoder pretrained on artificial data and contextualized word embeddings. It reaches an in-domain accuracy of 77% and an out-of-domain accuracy of 73%. Regarding the different error types, they conclude that their models ”do not show evidence of learning non-local dependencies related to agreement and questions, but do appear to acquire knowledge about basic subject-verb-object word order and verbal argument structure” (Warstadt et al., 2019).

3 Materials and methods

3.1 Data

Three data sources were used in this work. The main dataset is DaLAJ, a single-error derivation of the SweLL-gold corpus. In addition to this, sentences presenting correct samples from SweLL-gold and the COCTAILL corpus were used.

3.1.1 SweLL-gold

SweLL-gold is a subcorpus of the **Swe**dish **L**earner **L**anguage corpus, a collection of 502 pseudonymized, normalized, and correction annotated essays written by adult Swedish learners of beginner, intermediate, and advanced levels (Volodina et al., 2019). The tagset includes 35 error correction tags, including morphological, syntactical, orthographic, punctuation, and lexical ones as well as exceptions such as corrections made as a consequence to other corrections, corrections that do not fit into any of the categories, or markup of unintelligible strings. Rudebeck and Sundberg (2021) provide detailed information on correction annotation in the SweLL-gold data. The 502 SweLL-gold essays contain a total of

- 6,615 sentences containing one or more errors
- 1,706 correct sentences.

3.1.2 DaLAJ

DaLAJ is a single-error sentence-scrambled extension to the SweLL-gold corpus. The format is described in Volodina et al. (2021a), Volodina et al. (2021b), where the pilot version DaLAJ 1.0 was tested, based on four error types.[1] The full dataset used in our present study follows the same principles but contains 35 error types and therefore more sentence pairs. The basic principle of

[1]DaLAJ 1.0 is available as part of the SwedishGlue collection (https://spraakbanken.gu.se/en/resources/dalaj), while DaLAJ 2.0, the full version used for training and testing in this article, will be released at a later stage.

the DaLAJ format is that sentences that originally contained more than one error are included once for each error, with all other errors corrected. This has two advantages: Since larger parts of every sentence are correct, it is easier for the models to learn the patterns and structure of correct language than when sentences contain multiple errors. By splitting multi-error sentences into multiple single-error sentences, we obtain a DaLAJ version of the SweLL-gold corpus which is around five times bigger than the original SweLL-gold corpus. For every sentence, this dataset contains the wrong sentence, the corrected sentence, the pair of the wrong and correct tokens, and the error label as described above. In terms of metadata, it additionally has the education level of the course the student was taking when writing the text (split into beginner, intermediate, and advanced) (see Table 1). It also includes the student's first language, but this is not considered in the present work. The sentences are randomized, which excludes the possibility to reconstruct full essays. This way it is possible to avoid restrictions imposed by the GDPR (EU Commission, 2016).

Description	Example sentence
original sentence	§Den§ är en svår fråga .
corrected sentence	§Det§ är en svår fråga .
error-correction pair	§Den§–§Det§
error label	L-Ref
education level	Fortsättning

Table 1: Example sentence from DaLAJ

Here are a few statistics about the size and composition of DaLAJ 2.0 before preprocessing:

- Number of incorrect sentences: 26,652 with their corrected equivalents which represent 6,615 unique sentences
- Number of unique correct sentences: 6,615
- Number of tokens: 1,241,754
- Vocabulary size: 19,963

For effective model training, we need to have a balanced number of (unique) correct and incorrect sentences. However, as we can see from the statistics numbers, for the 26,652 sentences containing errors we have only 6,615 unique corrected sentences that are duplicated each time when a source original sentence has more than one error. To expose our models to sufficient number of correct sentences, we, therefore, ideally need to add further 20,000 correct sentences. 1,706 of those come from the SweLL-gold. To complement the rest, we use COCTAILL, a corpus of course books, as described below.

3.1.3 COCTAILL

COCTAILL was chosen as a source for the additional correct sentences because it comes from the realm of language learning and should therefore be similar in domain to DaLAJ. It also includes information about the level of the course at which the texts are used for teaching. We have, thus, a proficiency level label for each sentence in COCTAILL. We use this metadata to keep the original distribution of beginner (A-levels), intermediate (B-levels), and advanced (C-levels) sentences in the additional correct sentence.

COCTAILL stands for "**C**orpus **o**f **C**EFR-based **T**extbooks **a**s **I**nput for **L**earner **L**evel's modelling" and contains texts from 12 Swedish course books from beginner to advanced learners (Volodina et al., 2014). Since it also contains a fair amount of incomplete sentences such as headings, lists, or word definitions, we applied some filtering steps. In total, 5,015 beginner, 2,468 intermediate, and 5,066 advanced sentences were replaced with sentences of equivalent level to keep the original distribution.

3.2 Preprocessing

3.2.1 DaLAJ 2.0

We divided the DaLAJ sentences into three splits of 80% for training and 10% each for validation and testing, making sure that, even with duplicates, no identical sentences occur in the training and test splits and that the distribution of beginner, intermediate, and advanced sentences is equal across splits.

In the next step, we removed

- sentences with a length over 50 tokens (incl. punctuation)
- duplicate incorrect sentences
- all sentences that contained error types that appear less than 100 times in total (M-Other, M-Adj/adv, S-Comp, L-FL, S-Other, P-Sent, S-Adv, S-WO, S-FinV, S-R, P-R, S-Type) and

- all sentences that contained error types that do not belong to the five main error groups (orthography, lexis, morphology, punctuation, syntax) - i.e. tags that correspond to comments of all types and indicate illegible/uninterpretable strings (C, Cit-FL, Com!, OBS!, X)

Lastly, all pseudonymized tokens (e.g. 'A-city') were replaced with names of existing city, country, or place names, as shown in the example:

- Original: §jag§ är född i *A-hemland* .
- Replaced: §jag§ är född i *Norge* .

3.2.2 Training and validation sets

We tried two approaches with regards to data balance: (1) In the first approach, we kept the duplicate corrected sentences. Even though duplicates do not add new information to a model, they do keep it balanced, so it does not adopt bias due to an unequal label distribution. (2) In the second approach, we removed duplicates from the training and validation sets and replaced them with correct sentences from COCTAILL, as described in section 3.1.3.

3.2.3 Test sets

The models were evaluated on three different test sets. This does not just give insights into the models' performance but also into the impact the different compositions of the test sets have on the scores.

Test set 1 is the regular test split as it occurs in the dataset. In order to get accurate results, the correct sentences in this split were manually checked and corrected, so some changes were made, but no additional sentences were added or removed. This means that this test set contains a high number of duplicate correct sentences (as does the original dataset and the training and validation data in Models 1 and 3).

Test set 2 is a test set that includes no duplicates. It has the same number of incorrect sentences as the first test set and also uses the manually checked correct sentences. However, all duplicates were excluded, leaving this set significantly smaller and unbalanced.

In **test set 3**, we balanced test set 2 (the set without duplicates) by adding correct sentences from the original SweLL-gold corpus. These are not part of the DaLAJ training and validation sets, so they are unseen by the models, but come from the same domain as the other test sentences. One drawback here is that there are not enough intermediate-level sentences in the replacements, so they were supplemented with advanced-level sentences to make up for the difference. Table 2 gives an overview of all training, validation, and test sets.

3.3 Features

Different features were used in our models, alone or in combination, and with varying degrees of success. In all of them, we used white-space tokenization and padded to the maximum length of 50 with zeros on the left side of the sentence, unless otherwise specified.

FastText: First, words were converted into 300-dimensional pretrained FastText embeddings[2] (Grave et al., 2018). Pseudo-random vectors were used for infrequent words (UNK) and words that are not part of the embedding vocabulary (ERR). Missing words in the incorrect sentences were represented by "§§"-tokens. In the training and validation sets, they got the "UNK"-label and vector, in testing they got skipped, since adding them would have given away information about the error to the model.

FastText + error word: FastText embeddings like above were used, but with the error word explicitly added to the end of the sentence. For training and validation, we got the embeddings for the sentence as well as the error word(s) as described above and then concatenated the two vectors. For testing, the "ERR"-embeddings were added when out-of-vocabulary words occurred. Otherwise, only padding was added to the sentence embedding.

BERT: Contextualized word embeddings from Swedish BERT[3] (Malmsten et al., 2020) were used. A pretrained BERT-tokenizer split the sentences into words or subwords, which were then put through the pretrained Swedish BERT model. For the embeddings, we summed the hidden states of the last four encoder layers for each word. This resulted in 768-dimensional word embeddings. The BERT embeddings were padded on the right side to be compatible with the BERT tokenizer.

Word indices: Each word was simply converted to an index in the vocabulary and later turned into

[2]https://fasttext.cc/docs/en/crawl-vectors.html
[3]https://huggingface.co/KB/bert-base-swedish-cased

Set	# sen total	# beginner sen	# intermediate sen	# advanced sen	vocab
Train (dupl.)	32,394	12,890	5,766	13,738	10,826
Train (COCTAILL)	32,394	12,890	5,766	13,738	21,936
Val (dupl.)	4,008	1,576	722	1,710	2,922
Val (COCTAILL)	4,008	1,576	722	1,710	6,203
Test (dupl.)	3,884	1,439	659	1,786	2,677
Test (no dupl.)	2,573	1,001	437	1,135	2,677
Test (SweLL)	3,884	1,564	518	1,802	4,005

Table 2: Dataset and vocabulary sizes

100-dimensional embeddings by an Embedding layer[4] in the neural network. Words that occurred less than three times were regarded as unknown.

Character embeddings/indices: The sentences were converted into sequences of character indexes. They were transformed to 50-dimensional embeddings by an Embedding layer in the neural network. The threshold for unknown characters was set to five occurrences.

One-hot encodings for error words: Finally, one-hot vectors were used to indicate the problematic parts of each sentence. For training and validation, the word(s) between the §-markers were represented by 1, all other words and padding with 0. For testing, only words that do not occur in the FastText vocabulary were marked as 1 based on the assumption that these are spelling mistakes; all other words - as 0.

In addition to the word representations, we tried adding explicitly linguistic features, POS-tags and dependency relations. These tags were extracted with the Sparv pipeline[5] (Borin et al., 2016), converted into numbers by indexing the respective tags, and also padded to a length of 50 with zeros on the left side.

For the gold standard and for analysing the results, each sentence has two labels. One is the binary gold target indicating whether a result should be predicted as correct (0) or incorrect (1). The second is the SweLL error tag, indicating what exactly is wrong in the sentence. Correct sentences do not have an error tag.

The PyTorch Dataset and Dataloader classes[6] were used to shuffle and batch the data (batch size 32) and load it to the models.

[4]https://pytorch.org/docs/stable/generated/torch.nn.Embedding.html

[5]https://spraakbanken.gu.se/verktyg/sparv

[6]https://pytorch.org/docs/stable/data.html#torch.utils.data.DataLoader

3.4 Models

All models are based on a bidirectional LSTM layer and a linear layer. The choice of bi-LSTM classifier is based on its previous successful uses for binary error detection reported in literature (Rei and Yannakoudakis, 2016; Kaneko et al., 2017; Kasewa et al., 2018; Bell et al., 2019; Dekksne, 2019). BiLSTMs are useful for sequential data when long-distance dependencies also play a role and context on both sides of a token should be taken into account.

To get predictions from the output logits, softmax and argmax functions were used. The Adam optimizer was used with different learning rates. Loss was calculated with the Cross-Entropy Loss function. All models were trained for a maximum of 75 epochs with early stopping after 15 epochs without improvements in validation loss. The models differ in their specific hyperparameters, input, and structure. Many of the features and feature combinations did not give meaningful results or did not improve the results reached with simpler models. In the following, the successful models are described in more detail. For these, the respective results are discussed in section 4.

3.4.1 Model 1 & 2: FastText

The first two models took pretrained FastText embeddings as input with a hidden size of 100 and the learning rate 0.0001. Model 1 used the regular DaLAJ 2.0 data including duplicate correct sentences. Model 2 used the training and validation sets in which duplicate correct sentences were replaced by sentences from COCTAILL.

3.4.2 Model 3 & 4: BERT

Models 3 and 4 had the same basic structure but used contextualized BERT embeddings instead of FastText. The hidden size was 100, like in the models above, but the learning rate was reduced to 0.00005. As above, model 3 was trained using the

regular single-error dataset without additional sentences, while model 4 used additional COCTAILL sentences in training and validation.

3.4.3 Other models

Further experiments included adding linguistic features (such as parts of speech and dependency relations, character embeddings, word indices, one-hot encodings for error words) to test if they can improve the performance. They have in general failed compared to Models 1-4, and we therefore do not report them here, but outline in an Appendix.

4 Results and analysis

The models were evaluated in multiple ways. First, an overall quantitative analysis compared the different models. In the second and third part, the best-performing model was analyzed in more detail, considering error types and education levels. Finally, a qualitative analysis of the best models' predictions was conducted.

For the quantitative analysis, the main focus is on the accuracy score. However, since related work is often evaluated with other metrics such as F1-score, F0.5-score, or precision and recall, these scores are also reported for the best-performing model.

4.1 Overall quantitative analysis

There are three things to consider in the overall results in Table 3: The comparison between different embeddings, different training and validation sets, and between different test sets. First, regarding the embeddings, the models trained on BERT embeddings (Model 3 and Model 4) clearly outperformed the ones trained on FastText across all combinations of training and test sets. Second, the highest score (for both embedding types) was reached on models trained and validated on the dataset where duplicates were replaced with sentences from COCTAILL. Third, the differences between test sets show that models performed better on test sets without duplicates. This pattern was not as clear in the models trained on data including duplicates.

Table 4 contains the full classification report for the best model on the best test set. A look into these more detailed results shows significantly higher precision, recall, and F-scores for the incorrect sentences than the correct ones. This indicates that the model learned more from the incorrect than the correct samples in the training, potentially because there is more variation in the incorrect sentences. A comparison between the individual scores shows very stable results. Within the two classes, precision and recall lie very close together. In binary classification, there is usually a certain trade-off between precision and recall, and which one is more important depends on the task and application. Our model here turned out to be very balanced in this regard, so the F1- and F0.5-scores are almost identical.

4.2 Performance by error type

For all further analysis, only Model 4, which has the best overall performance, is considered. The following results are taken from test set 2.

Due to our filtering and preprocessing steps, we used only 18 of the total 35 SweLL error types in our experiments. Table 5 shows the accuracy and number of samples in the test set for each of them, along with a short explanation of the types. For a table explaining all error types we refer the reader to the appendix of Volodina et al. (2021a). More detailed information can be found in the full correction annotation guidelines[7] (Rudebeck and Sundberg, 2021). This only takes the incorrect sentences into account, since the correct ones do not have an error type. Both the individual scores and the ranking of error types differed between different models. Therefore, the following observations only allow conclusions about this specific model.

First, the types with the highest accuracy are considered. Some of them are expected. *O, L-Der,* and *M-F*[8], for example, are types that often result in "words" that do not exist in correct Swedish and are thus not part of the word embeddings used to train the models. Other high-performing groups were more surprising. The high scores for *S-Clause, S-Ext*, and *S-Msubj* indicate that the model learns about more complex aspects of language, such as word order. The fact that *P-W* errors are among the most successful groups further supports the conclusion that this model has a decent understanding of Swedish sentence structure.

Second, *O-Comp* and *M-Num* are the types with the lowest accuracy in this model. *O-Comp* might be more difficult to predict than other errors since this aspect of a language often does not follow

[7] https://spraakbanken.github.io/swell-project/Correction-annotation_guidelines

[8] All correction codes are briefly explained in Table 5

Model	data	embeddings	test 1 (dupl.)	test 2 (no dupl.)	test 3 (SweLL)
1	DaLAJ	FastText	0.61	0.42	0.53
2	DaLAJ + COCTAILL	FastText	0.59	0.62	0.66
3	DaLAJ	BERT	0.66	0.67	0.65
4	DaLAJ + COCTAILL	BERT	0.61	0.73	0.69

Table 3: Accuracy of models 1-4 in three different test sets

Class	Precision	Recall	F1-score	F0.5-score	Sample number	Accuracy
0 (correct)	0.43	0.39	0.41	0.42	631	0.39
1 (incorrect)	0.81	0.83	0.82	0.81	1942	0.83
Total					**2573**	**0.73**

Table 4: Classification report for model 4 on test set 2

strict rules. For *M-Num* errors, there might be difficulties in learning longer-distance agreement when determiners, nouns, and adjectives are not directly adjacent. However, it is somewhat surprising that the related errors *M-Def* and *M-Gend* perform significantly better.

Model's performance by error group does not show a very clear pattern. Most groups include mixed success rates across their respective types. That being said, lexical and punctuation errors are generally closer to average, while morphological errors tend to perform lower and syntactical ones perform above average.

A last perspective for comparison is the number of samples of each type in the dataset. One might expect a strong positive correlation between number of samples and prediction accuracy of an error type. However, this was not quite the case here. It is true that the error types with low accuracy scores generally also have a low number of samples (e.g. *O-Comp*). This pattern does not hold for the entire set of results though, since some of the types with very high accuracy, such as *S-Ext* or *S-Msubj*, also have relatively low number of samples. Finally, *P-M* and *S-M* are two types with above-average sample sizes, but merely average accuracy scores, indicating that identifying missing tokens in a sentence might be inherently more difficult than identifying incorrect ones.

4.3 Performance by education level

Table 6 shows clear performance differences between sentences written by learners at different education levels. Beginner sentences are predicted with distinctly higher success than intermediate and advanced ones. This might partly be explained by the under-representation of intermediate-level sentences. Another reason is the unequal distribution of error types across levels. Some of the types that proved to be most successful in the section above, such as *O, O-Cap*, or *M-F* occur with higher frequency in the beginner set. At the same time, some of the overall less successful types, such as *M-Case, M-Num*, or *L-W*, occur more frequently in the sentences written by advanced learners.

4.4 Qualitative analysis

In this section we take a closer look at the predictions, especially the false negatives, of Model 4. Numbered example sentences can be found at the end of the section.

First, there are small issues in the dataset. Some sentences were apparently incorrect when annotated in the context of their text, but are correct when considered independently. Example [1] is one case which the model therefore "misclassifies" as correct. Another problem is that some sentences have essay titles or headings incorrectly attached to them, like in [2].

Apart from these issues, there are some specific errors the model frequently misses. One of them is agreement with longer distances between the respective words, for example in [3]. Another difficulty for the model seem to be preposition choices. Incorrect usage of for example *"i", "på", "för"*, or *"med"* is often not predicted as an error. Sentences in which the pronoun case is incorrect also appear frequently among the false negatives. One last group of errors that are not recognized well by the model are spelling mistakes in names.

One step in preprocessing, the naive replacement of pseudonymization tokens with city, country, or place names, resulted in some sentences of

Error tag	Explanation	# sen	# true	Acc
O-Comp	Orthography: Problem with compounding	18	13	0.72
O-Cap	Orthography: Wrong capitalization	29	25	0.86
O	Orthography: Regular spelling correction	261	235	0.90
L-Der	Lexical: Word formation problem (derivation or compounding)	58	49	0.84
L-Ref	Lexical: Choice of anaphoric expression	59	48	0.81
L-W	Lexical: Wrong word or phrase	319	257	0.81
M-Case	Morphology: Noun case correction (nom vs gen; nom vs acc)	31	24	0.77
M-Def	Morphology: Definiteness (articles; noun & adj forms)	280	222	0.79
M-F	Morphology: Grammatical category kept, form changed	24	21	0.88
M-Gend	Morphology: Gender correction	81	67	0.83
M-Num	Morphology: Number correction	81	59	0.73
M-Verb	Morphology: Verb corrections (inflections, auxiliaries)	202	173	0.86
P-M	Punctuation: Punctuation missing (added)	134	113	0.84
P-W	Punctuation: Wrong punctuation	38	33	0.87
S-Clause	Syntax: Change of clause structure, incl. synt. function	66	61	0.92
S-Ext	Syntax: Extensive and complex correction	26	24	0.92
S-M	Syntax: Word missing (added)	196	157	0.80
S-Msubj	Syntax: Subject missing (added)	39	36	0.92

Table 5: Accuracy and number of samples by error type (in the test set) in Model 4

Education level	# Samples	Accuracy
Beginner	1001	0.77
Intermediate	437	0.69
Advanced	1135	0.70

Table 6: Accuracy and number of samples (in the test set) by education level in Model 4

questionable logic, like [4]. Looking at the results, it does not seem to disturb the classifier, but more research into it would be needed to be sure. Finally, we found a pattern that longer sentences tend to get predicted as incorrect more often than shorter ones. This is not conclusive by itself but invites further research into the effect of sentence length on the models.

[1] *Jag §är§ väldigt bra .*

[Eng. I §am§ very good .]

[2] *Skrivuppgift 3 , 3 april 2018 Politiker som föredömen Får politiker vara §gott§ föredömen för medborgarna ?*

[Eng. Writing task 3 , 3 April 2018 Politicians as models Are politicians allowed to be §good§ models for citizens ?]

[3] *Han har svart hår , mörka ögon och en mun som alltid §ville§ skratta .*

[Eng. He has black hair , dark eyes and a mouth that always §wanted§ to smile .]

[4] *Ruinen ligger mellan Spanien och Danmark och §den§ hade inte tak §utan§ bara fyra väggar .*

[Eng. The ruins lie between Spain and Denmark and §it§ has no roof §but§ only four walls .]

5 Discussion

The first conclusion to be drawn from the results is that there are significant differences in the effectiveness of different types of word embeddings. The fact that the models trained on BERT embeddings perform higher than the ones trained on FastText across all combinations of training and test sets presents them as the better choice overall. Reasons for this could be the differences in training data, dimensionality, and that the method of getting embeddings from the context itself works better in this task.

Our second insight is that there are clear differences in how successfully each error type is predicted. These differences are only partially correlated with the types' representation in the training data. As a general tendency, spelling mistakes and simple word-order errors are predicted with exceptionally high success rates while morphological errors (especially agreement of non-adjacent words) perform worse. These trends have to be taken with caution, however. Some error types occur in very few samples in the test set, which might impact the score's reliability in these cases.

Furthermore, we found that there are differences in performance depending on the sentences' education level. Sentences written on the beginner level proved to be classified with significantly higher success than those on the intermediate and advanced level. One explanation could be the under-representation of intermediate-level sentences. Another one is that the distribution of error types is not equal across the proficiency levels.

A comparison with similar studies on English shows that our work lies well within the range of their results. For example, in the AESW 2016 task, teams reached F1-scores of up to 62% on sentence-level classification of scientific writing (Daudaravicius et al., 2016). Warstadt et al. (2019) reached 73% to 77% accuracy on their CoLA dataset. Their data consists of sentences that were purposefully written to illustrate certain errors and that are not originally embedded in the context of a text, which is a big difference to the DaLAJ data.

The fact that our results compare favorably to similar studies in English proves that the novel approach used to create DaLAJ dataset was successful. As explained in more detail in Volodina et al. (2021a), there are several advantages to using a dataset based on learner data for this task. Not only is the data realistic, it is also generally annotated by experts, and often includes detailed error labels. Advantages of the hybrid approach between authentic and synthetic data are that the number of available sentences is higher with this method, sentences are more informative than authentic ones, but still very similar to the originals. A minor drawback of this dataset is that the sentences were originally written and normalized (i.e. re-written in correct Swedish) in the context of a full essay and then classified in isolation, which caused some difficulties with predicting the correctness of for example anaphoric references.

The experiments with different training, validation, and test sets gave a clear indication that replacing duplicate sentences with unique ones from another source results in better models and better scores. By replacing the duplicates with correct sentences from a second corpus, they have far more relevant input and are able to generalize better.

6 Conclusions and future work

We presented promising benchmark results on the linguistic acceptability task in Swedish. The comparison of different input features showed that pretrained word embeddings, especially contextualized BERT embeddings, are very successful while other ways of representing the sentences did not yield good results, and additional linguistic features did not improve the embedding-based model. Overall, the dataset proved to be big and informative enough to train such models, despite some minor drawbacks.

In future experiments, we plan to use this dataset for multi-class classification of errors, for token-level error detection, and for error correction. These experiments would be an important step towards a functioning automatic writing evaluation (AWE) system for Swedish, where feedback generation will need to rely on correctly detected and labeled error types. In connection to this, we will need to see whether models trained on distilled hybrid data like DaLAJ can be successfully applied to authentic data containing multiple errors per sentence. Finally, we will experiment with generation of synthetic data to study its influence over model performance and to improve our chances of getting accurate tools for language learners.

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

Samuel Bell, Helen Yannakoudakis, and Marek Rei. 2019. Context is key: Grammatical error detection with contextual word representations. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 103–115, Florence, Italy. Association for Computational Linguistics.

Lars Borin, Markus Forsberg, Martin Hammarstedt, Dan Rosén, Roland Schäfer, and Anne Schumacher. 2016. Sparv: Språkbanken's corpus annotation pipeline infrastructure. In *SLTC 2016. The Sixth Swedish Language Technology Conference*, pages 17–18, Umeå University, Sweden.

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.

Daiga Deksne. 2019. Bidirectional LSTM tagger for Latvian Grammatical Error Detection. In *International Conference on Text, Speech, and Dialogue*, pages 58–68. Springer.

EU Commission. 2016. General data protection regulation. *Official Journal of the European Union*, 59:1–88.

Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomas Mikolov. 2018. Learning word vectors for 157 languages. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*.

Wang Jing, Matthew A. Kelly, and David Reitter. 2019. Do we need neural models to explain human judgments of acceptability? *CoRR*, abs/1909.08663v1.

Masahiro Kaneko, Yuya Sakaizawa, and Mamoru Komachi. 2017. Grammatical error detection using error-and grammaticality-specific word embeddings. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 40–48.

Sudhanshu Kasewa, Pontus Stenetorp, and Sebastian Riedel. 2018. Wronging a right: Generating better errors to improve grammatical error detection. *CoRR, abs/1810.00668*.

Jey Han Lau, Alexander Clark, and Shalom Lappin. 2017. Grammaticality, acceptability, and probability: A probabilistic view of linguistic knowledge. *Cognitive Science*, 41(5):1202–1241.

Steve Lawrence, C. Lee Giles, and Sandiway Fong. 2000. Natural language grammatical inference with recurrent neural networks. *IEEE Transactions on Knowledge and Data Engineering*, 12(1):126–140.

Lung-Hao Lee, Bo-Lin Lin, Liang-Chih Yu, and Yuen-Hsien Tseng. 2016. The NTNU-YZU system in the AESW shared task: Automated evaluation of scientific writing using a convolutional neural network. In *Proceedings of the 11th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 122–129, San Diego, CA. Association for Computational Linguistics.

Tal Linzen, Emmanuel Dupoux, and Yoav Goldberg. 2016. Assessing the ability of LSTMs to learn syntax-sensitive dependencies. *Transactions of the Association for Computational Linguistics*, 4:521–535.

Martin Malmsten, Love Börjeson, and Chris Haffenden. 2020. Playing with words at the national library of sweden - making a swedish BERT. *CoRR*, abs/2007.01658v1.

Rebecca Marvin and Tal Linzen. 2018. Targeted syntactic evaluation of language models. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1192–1202, Brussels, Belgium. Association for Computational Linguistics.

Marek Rei and Helen Yannakoudakis. 2016. Compositional sequence labeling models for error detection in learner writing. *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*.

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.

Allen Schmaltz, Yoon Kim, Alexander M. Rush, and Stuart Shieber. 2016. Sentence-level grammatical error identification as sequence-to-sequence correction. In *Proceedings of the 11th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 242–251, San Diego, CA. Association for Computational Linguistics.

Carson T. Schütze. 1996. *The Empirical Base of Linguistics : Grammaticality Judgments and Linguistic Methodology*. University of Chicago Press, Chicago, Il.

Ekaterina Taktasheva, Vladislav Mikhailov, and Ekaterina Artemova. 2021. Shaking syntactic trees on the sesame street: Multilingual probing with controllable perturbations. *arXiv preprint arXiv:2109.14017*.

Elena Volodina, Lena Granstedt, Arild Matsson, Beáta Megyesi, Ildikó Pilán, Julia Prentice [Grosse], Dan Rosén, Lisa Rudebeck, Carl-Johan Schenström, Gunlög Sundberg, and Mats Wirén. 2019. The SweLL language learner corpus: From design to annotation. *The Northern European Journal of Language Technology*, 6:67–104.

Elena Volodina, Yousuf Ali Mohammed, and Julia Klezl. 2021a. DaLAJ - a dataset for linguistic acceptability judgments for Swedish. In *Proceedings of the 10th Workshop on Natural Language Processing for Computer Assisted Language Learning (NLP4CALL 2021)*, pages 28–37.

Elena Volodina, Yousuf Ali Mohammed, and Julia Klezl. 2021b. DaLAJ - a dataset for linguistic acceptability judgments for Swedish: Format, baseline, sharing. *CoRR*, abs/2105.06681.

Elena Volodina, Ildikó Pilán, Stian Rødven Eide, and Hannes Heidarsson. 2014. You get what you annotate: A pedagogically annotated corpus of coursebooks for Swedish as a second language. In

Proceedings of the third workshop on NLP for computer-assisted language learning, pages 128–144, Uppsala, Sweden. LiU Electronic Press.

Alex Warstadt, Amanpreet Singh, and Samuel R. Bowman. 2019. Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics*, 7:625–641.

Appendix A. Failed experiments

Character embeddings/indices: Since words with orthographic and morphological errors often do not occur in the word embeddings used, we hypothesized that character-level representations might be better-suited. Therefore, we trained a model on character instead of word embeddings. Apart from that, it had the same structure as the models with pretrained word embeddings. This model performed better than chance, but clearly worse than the FastText and BERT models, reaching accuracies of 55% to 64%. A possible reason for the low performance is the relatively low amount of data for training embeddings. Future approaches might be to separately train a character-level language model on a bigger correct dataset and use that for the embeddings or to try other methods of capturing subword information, such as byte-pair encodings.

Word indices: The next experiment used the same index-based approach, but on the word level again. Since the word embeddings used in this experiment are trained on very different data (Wikipedia, newspaper articles, etc.) than learners' essays, we tried using in-domain embeddings. Similar to the model above, it reached accuracy scores of 52% to 60%, possibly also due to the comparatively small dataset.

FastText + error word: We had two reasons for adding the error word to the FastText embeddings. First, it introduced more variety among the correct sentences in the models with duplicates. Second, repeating the wrong word could have helped the model learn what exactly is wrong in a sentence. This model did reach higher validation accuracy (up to around 70%), but accuracy on the test set remained at or around 50%. This indicates that the additional information is useful to the model to some extent, but it cannot transfer that knowledge to sentences where the error word is not explicitly repeated.

One-hot encodings for error words: This feature was again combined with the pretrained word embeddings. Both input vectors went through separate biLSTM layers, and the outputs were concatenated before the linear layer. Validation accuracy improved, but not test accuracy, so the problem seems to lie in the transfer of information to the test sentences, which mainly consist of only zeros (except for spelling errors). An idea for improving this is to randomly replace the one-hot vectors for some sentences in the training and validation data with zeros-only vectors, forcing the model to generalize to data with only zeros. Another approach might be to use a more advanced model with an attention mechanism instead.