Generation of Synthetic Error Data of Verb Order Errors for Swedish

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

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

Abstract

We report on our work-in-progress to generate a synthetic error dataset for Swedish by replicating errors observed in the authentic error-annotated dataset. We analyze a small subset of authentic errors, capture regular patterns based on parts of speech, and design a set of rules to corrupt new data. We explore the approach and identify its capabilities, advantages and limitations as a way to enrich the existing collection of error-annotated data. This work focuses on word order errors, specifically those involving the placement of finite verbs in a sentence.

1 Introduction

The lack of sufficient data to train algorithms capable of detecting, labeling and correcting grammatical errors calls for the need to generate synthetic (i.e. machine-made, not human-produced) error datasets to enrich the existing resources. As mentioned by Stahlberg and Kumar (2021), the need for synthetic datasets (aka corrupt or artificial datasets) exists not only for low-resource languages, but also for high-resource languages like English. This is due to the fact that data for error detection and correction is far more sparse than required for most tasks in NLP, as grammatical errors are found in different frequencies and distributed unevenly across written language. Moreover, the appearance of grammatical errors in student essays depend notably on the speaker's particularities, such as their proficiency level, native language(s) and age. The need is especially acute for languages that are on the low-resource end in this respect, as is the case for Swedish.

In this paper, we present a pilot study to generate artificial error data for Swedish by mimicking error patterns present in authentic error datasets, namely, in the SweLL learner corpus (Volodina et al., 2019) and its one-error-per-sentence DaLAJ derivative (Volodina et al., 2021). We create a corruption pipeline to insert artificial errors into the sentences from COCTAILL, a corpus of textbooks used for teaching Swedish (Volodina et al., 2014). We expect the artificially produced error data to be a valuable resource for such tasks as Grammatical Error Detection / Labeling (GED) and Grammatical Error Correction (GEC) for Swedish, which at the moment are dormant fields.

In this pilot, we focus on word order errors involving placement of finite verbs (tagged S-FinV). The final dataset comprises 31,788 corrupted sentences each containing one error of the syntactical error type "S-FinV", paired with their correct counterparts. The code and the generated data can be found on GitHub[1].

2 Related work

Recently much attention has been given to practical and theoretical aspects of artificial error data generation as a way to enhance performance of grammatical error correction systems, both with respect to methods of generation, source (aka seed) corpora used for corruption and the ways pseudo-data is used in system architectures (e.g. Flachs et al., 2021). Takahashi et al. (2020) give probably the most nuanced introduction to the problem.

Approaches to generation of synthetic error datasets can be roughly divided into rule-based and model-based ones, which further exhibit variation with regards to presence or absense of error labels. Advantages of *model-based approaches* (e.g. Stahlberg and Kumar, 2021) is that they capture the variety of error types present in the authentic data and the artificial data is fast to generate. However, training a model for replicating errors requires access to large amounts of such data, which often is a problem to start with. It has also been observed that models may show biases towards the data they have been trained on, with a consequence that they are

[1] https://github.com/juditcasademont/Generation-of-synthetic-error-data-LTR-project

Proceedings of the 17th Workshop on Innovative Use of NLP for Building Educational Applications (BEA 2022), pages 33 - 38

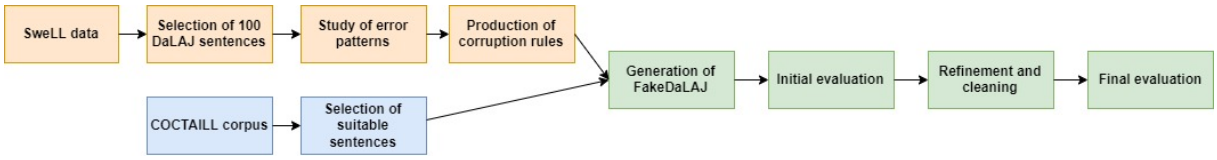

Figure 1: Overview of the pseudo-data generation process.

not general enough for unseen contexts or domains (e.g. Bryant et al., 2017).

Advantages of *rule-based approaches* (e.g. Grundkiewicz et al., 2019) are directly opposite, namely, that they can be created with zero or minimal access to gold data and can better generalize since they are kept on an abstract level. Some known rules include simple random operations, e.g. deletion of a word, randomly swapping neighbouring words, exchange of one inflected form with another or use of so-called confusion pairs, i.e. incorrect segment/token > corrected variant (e.g. Choe et al., 2019; Grundkiewicz et al., 2019).

A more linguistic approach to error rules, e.g. through abstracting to part of speech (POS) patterns or patterns including morpho-syntactic information, requires more time for designing corruption rules, but has one obvious advantage: using such rules allows control over the generated data and, importantly, it is possible to add error labels to corrupted sentences, which makes the pseudo-data applicable both to error correction and to error classification tasks. It also has an advantage of inserting realistic errors typical of learners, and has been shown to increase performance of GEC systems, compared to random error types (Takahashi et al., 2020) . Given the scarcity of the Swedish authentic data, we experiment with rule-based approaches using linguistic analysis to extract typical error patterns and to generate synthetic errors based on those.

3 Data and resources

The overview of the pseudo-data generation is shown in Figure 1. On the left, the top level shows steps for working with learner data, starting from the SweLL data, including preparation of one-error-per-sentence DaLAJ dataset, analysis of a 100 cleaned DaLAJ samples for identification of error patterns, and production of corruption rules. The bottom level on the left shows a parallel work involving selection of a seed corpus with correct language and preselection of sentences for corruption. On the right, the graph shows the process for corruption of the seed data and its subsequent cleaning and evaluation.

To perform the task at hand, thus, two main sources of textual data are needed: a corpus of tagged errors and a corpus of clean (i.e., error-free) texts. Additionally, a part of speech (POS) tagger to extract grammatical information is required.

3.1 Error-labeled learner data

In this project, we use SweLL-gold (Volodina et al., 2019), a collection consisting of 502 learner texts, manually corrected and tagged according to 6 top error categories which, in turn, have their own sub-categories (Rudebeck and Sundberg, 2021). The top error types are: **O**rthographic, **L**exical, **M**orphological, **P**unctuation, **S**yntactical and Other (the category Other contains comments and unitelligible strings). All texts are original and each sentence on average contains more than one error of more than one kind. The focus of this project is on **S**yntactical errors involving the position of **Fin**ite **V**erbs in a sentence, tagged in the SweLL corpus as “S-FinV". There are 701 instances of this tag in the corpus.

To better represent error types, we convert SweLL-gold to DaLAJ format (Volodina et al., 2021) where SweLL-gold data is represented as a set of sentence pairs (original-corrected) in a scrambled order. The most attractive feature of DaLAJ is that each sentence contains one error of one type only. This means that an original SweLL sentence has as many instances in the DaLAJ dataset as there are individual correction tags in its original form. This format supports easier detection and analysis of error patterns, for both humans and computers.

3.2 Seed data

The source of the error-free data, i.e. seed data to be corrupted with automatically generated errors, is the COCTAILL corpus (Volodina et al., 2014) containing 25,960 scrambled sentences from twelve course books of Swedish as a second language, labeled for levels of proficiency. They represent the following CEFR levels: Beginner (A1), Elementary (A2), Intermediate (B1), Upper Intermediate

(B2) and Advanced (C1). The Proficiency level (C2) is not represented. We assume that the lexical, grammatical and syntactical patterns in COCTAILL texts would be relatively close to the ones used in learner essays, thus fitting perfectly for our purposes. Out of the 25,960 sentences present in the COCTAILL corpus texts, 20,307 were deemed useful, as a filtering process was carried out to discard sentences not containing verbs as well as sentences shorter than two tokens.

3.3 POS tagging pipeline

Språkbanken Text's Sparv pipeline[2] (Borin et al., 2016) was used to extract grammatical information in the form of morphosyntactic tags. This pipeline was used in two distinct phases of the project: in the analysis of the error patterns and in the generation of corrupted data. The Sparv pipeline is a tool for text analysis that can be run from the command line or called programmatically through an API.

4 Methods

4.1 Error patterns

Swedish is a so-called "verb-second" language, which means that finite verbs, with a few exceptions, take the second position in a sentence (where positions are counted in phrases). Errors with placement of finite verbs are considered among the most typical ones for L2 learners of Swedish. Linguistic analysis of approx. 100 DaLAJ sentence pairs containing S-FinV errors has shown that three POS in specific positions in the sentence, have a tendency to be the cause of S-FinV errors, namely: pronouns (PN), nouns (NN) and adverbs (AB) (in the order of frequency). Additionally, there is a need to make a special case for proper names (PM).

Pronouns in the studied dataset are the most fruitful part of speech tag in the production of verb order errors, making two thirds of all S-FinV errors. The error production patterns involving pronouns can be grouped into two distinct groups: PN-VB →VB-PN and VB-PN →PN-VB (where the first part is correct →the second is erroneous).

To exemplify, in PN-VB →VB-PN*[3] errors, the error tends to happen right after a conjunction (KN), an interrogative or relative adverb (HA), or at the beginning of a sentence, like in the example below:

Jag **heter** Karin.[4] [5] →**Heter** jag Karin.*
Eng: My name is Karin.

The VB-PN →PN-VB* pattern, is decidedly the most frequent one in the "pronoun"-subtype, and appears in subordinate clauses, which requires the reversal of pronoun and verb positions. This phenomenon usually appears after interrogative or relative pronouns (HP) and adverbs (AB).

Errors involving the positions of verbs in relation to **adverbs** are also well-represented in our dataset, even though not as frequent as pronoun-related errors. Their typical error production patterns are: VB-AB →AB-VB* and AB-VB →VB-AB*.

In VB-AB →AB-VB* errors, the learner writes the adverb before the verb when its correct position is after the verb. It usually occurs in a sentence's main clause, probably because the writer wrongly applies the rule for subordinate clauses.

In contrast, errors of type AB-VB→VB-AB* appear in subordinate clauses where the verb and the adverb must switch positions in the sentence:

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

Error patterns involving **nouns** in close relation to verbs are slightly more varied than those having to do with pronouns and adverbs. The reason is that nouns can be modified by other parts of speech, such as determiners, possessives and adjectives. They can in addition be modified by adjective-like subordinate clauses.

Within this category, the primary error pattern is VB-NN →NN-VB* (or rather noun phrases), in which the verb needs to be placed before an unmodified noun. These errors are likely to occur when the initial position in a clause is taken by another word class, most frequently by an adverb:

Ibland **kommer** mormor. →
Ibland mormor **kommer**.*
Eng: Grandma comes sometimes.

Other subtypes involve pre-modifiers, e.g. determiners (DT), possessives (PS), adjectives (JJ):

(1) VB-DT-NN →DT-NN-VB;
(2) VB-PS-NN →PS-NN-VB, and
(3) VB-JJ-NN →JJ-NN-VB.

[2] spraakbanken.gu.se/sparv

[3] We use asterisk (*) to mark the incorrect pattern/example sentence

[4] 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.

[5] All examples, unless stated otherwise, belong to the SweLL and DaLAJ datasets.

Corrupted sentence	Seed sentence	Error index corrupted	Error index seed	Confusion pairs	Error label	Split
Det ungefär finns 5 000 språk i världen .	Det finns ungefär 5 000 språk i världen .	['s1', 's2']	['t2', 't1']	['ungefär', 'finns', '-->', 'finns', 'ungefär']	S-FinV	Train
Far : Men nu jag är jättehungrig !	Far : Men nu är jag jättehungrig !	['s4', 's5']	['t5', 't4']	['jag', 'är', '-->', 'är', 'jag']	S-FinV	Train
Det snö är i luften .	Det är snö i luften .	['s1', 's2']	['t2', 't1']	['snö', 'är', '-->', 'är', 'snö']	S-FinV	Train

Figure 2: Corrupted data, selected columns.

The final pattern is based on **proper names**, exhibiting similar behaviour to noun-based error patterns. Due to pseudonymization, pseudonyms are used instead of the originally used proper names (as in the example below).

Han visste inte om Brad Pitt **vann** priset. →
Han visste inte om **vann** Brad Pitt priset.*
Eng: He didn't know if Brad Pitt won the prize.

The typical patterns are: (1) PM-VB →VB-PM and (2) PM-PM-VB →VB-PM-PM.

4.2 Corruption method

Using the identified error patterns, we reverse them to a set of rules for each error subtype (pronouns, adverbs, nouns and proper names) for shifting the position of words in COCTAILL sentences. We first extract POS tags from the correct sentences and store them. In the process, sentences shorter than two tokens and those not containing verbs are discarded. All of them share an initial filter to avoid changing the position of words before a colon, in case a verb is present, like in the example below. Capitalization is toggled if the initial capitalized word is involved in the corruption.
Stryka subjektet: Jag **är** mycket trött. →
Stryka subjektet: **Är** jag mycket trött.*
Eng: Cross out the subject: I am very tired.

We strictly keep to the rule of having one error per sentence. However, sentences may appear more than once in the synthetic dataset, as they can be corrupted several times, for example, if sentences contain more than one verb or fit into several error sub-patterns. In the end, a final scramble is performed to the order of the sentences before they are stored in a .csv file, with suggestions for data splits (80%-10%-10%) and confusion pairs (Figure 2).

5 Results

A total of 31,788 sentences were corrupted from the 20,307 usable sentences available. The distribution of error sub-types is shown in Table 1:

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

Table 1: Error count of the final corrupted data.

Similarly to the frequency distribution in student essays, pronoun-dependent verb order errors are the most frequent ones in the corrupted data, with 41.05% of synthetic errors being of this type. The second most productive rules are the ones involving adverbs, with 31,21% of errors, followed by nouns at 25,3%. Finally, as expected, the corruption pipeline produced a considerably lesser quantity of errors involving proper names at 2,44%. The distribution in the corrupted data, thus, reflects the observed tendency in the authentic data.

To assess the quality of the corruption method, we carried out a small-scale evaluation. Two people have independently checked 100 randomly selected corrupted sentences in terms of how similar they are to hypothetical learner-made errors (i.e. to make sure they are high quality). Following Bryant et al. (2017), we used a three level scale of assessment: *Good*, *Acceptable* and *Bad*. For *Acceptable* and *Bad*, a reason could be indicated for further analysis.

The evaluation shows that 76% (67%) of sentences are *Good*, 14% (25%) are *Acceptable* and 10% (8%) are *Bad*. The numbers in brackets come from the second annotator. Some observed problems had to do with more complex phrase shifts that were missed. In others, the problem comes from the source data, incl. unfinished sentences with an uncertain sentence type (affirmative vs interrogative), which then sounds correct even if the verb and noun change places. It should be noted that the main purpose of this evaluation was to see whether humans think that the synthetic data will be useful for training algorithms, and the result where on average 90% sentences are either *Good* or *Acceptable* is very encouraging. It has been earlier claimed that even unrealistic errors are use-

Data type	Model type	Lexical	Morphological	Orthographic	Punctuation	Syntactical
Original learner data	BERT Bi-LSTM	0.54894179	0.60539215	0.57565789	0.46072507	0.64680232
Original learner data + 500 FakeDaLAJ	BERT Bi-LSTM	**0.60634328**	**0.63834422**	**0.61026936**	**0.56034482**	0.69732297
Original learner data + 1500 FakeDaLAJ	BERT Bi-LSTM	0.51798561	0.58823529	0.50641940	0.37499999	**0.71934945**

Table 2: F0.5 score results from some selected models on error classification task.

ful for pre-training GEC models (e.g. Flachs et al., 2021; Grundkiewicz et al., 2019). Given our results, therefore, we consider the produced dataset appropriate for the task.

We have run the first experiments exploring *effects of pseudo-data on the model performance* for the task of error detection and classification, where classification is limited to the top error categories (**O**rthographic, **L**exical, **M**orphological, **P**unctuation, **S**yntactical). Detailed description of that experiment is the topic of another publication, however, we can shortly name here that we have observed a tangible improvement of the classification results when 500 FakeDaLAJ sentences of S-FinV nature were added to the training data. When more sentences were added, the models seemed to learn to classify syntactical errors disadvantaging other error types. A sample of the results obtained, measured with the F0.5 score, are shown in Table 2.

6 Conclusions and future work

This paper introduces a process for generation of synthetic error datasets with corresponding error labels based on linguistic analysis of real-life learner errors in the context of limited error-annotated learner data. This process could be replicated for other error tags, or extended and adapted to other low-resource languages. Manually studying and designing corruption rules is time-consuming and can be inaccurate due to human error and language biases. Therefore, an alternative to optimize time and avoid human mistakes could be to rely on guided models as suggested by Stahlberg and Kumar (2021) or Sennrich et al. (2016). However, we have to adhere to rule-based approaches due to the lack of sufficient amount of gold data. Yet, we foresee considerable benefits of generating realistic errors.

The resulting fakeDaLAJ (S-FinV) dataset is released for public use.[6] Currently, we are testing this dataset in a task for error detection and classification. In the near future, we will also release a set of cleaned 100 DaLAJ sentences per each error tag in the SweLL-gold data, so that the community of interested researchers and developers can use them for generation of synthetic datasets for other error types.

[6]https://github.com/juditcasademont/Generation-of-synthetic-error-data-LTR-project

References

Lars Borin, Markus Forsberg, Martin Hammarstedt, Dan Rosén, Roland Schäfer, and Anne Schumacher. 2016. Sparv: Språkbanken's corpus annotation pipeline infrastructure. In *Proceedings of Swedish Language Technology Conference (SLTC)*. Umeå University.

Christopher Bryant, Mariano Felice, and Edward Briscoe. 2017. Automatic annotation and evaluation of error types for grammatical error correction. Association for Computational Linguistics.

Yo Joong Choe, Jiyeon Ham, Kyubyong Park, and Yeoil Yoon. 2019. A neural grammatical error correction system built on better pre-training and sequential transfer learning. *arXiv preprint arXiv:1907.01256*.

Simon Flachs, Felix Stahlberg, and Shankar Kumar. 2021. Data strategies for low-resource grammatical error correction. In *Proceedings of the 16th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 117–122.

Roman Grundkiewicz, Marcin Junczys-Dowmunt, and Kenneth Heafield. 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*, pages 252–263.

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

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data. *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 86–96.*

Felix Stahlberg and Shankar Kumar. 2021. Synthetic data generation for grammatical error correction with tagged corruption models. *CoRR*, abs/2105.13318.

Yujin Takahashi, Satoru Katsumata, and Mamoru Komachi. 2020. Grammatical error correction using pseudo learner corpus considering learner's error tendency. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 27–32.

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*.

Elena Volodina, Yousuf Ali Mohammed, and Julia Klezl. 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*.

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. *Proceedings of the third workshop on NLP for computer-assisted language learning*.