Error syntax aware augmentation of feedback comment generation dataset

Nikolay Babakov[1]*, **Maria Lysyuk**[2], **Alexander Shvets**[3], **Lilya Kazakova**[4], **and Alexander Panchenko**[2,5]
[1] Centro Singular de Investigación en Tecnoloxías Intelixentes (CiTIUS), Universidade de Santiago de Compostela
[2] Skolkovo Institute of Science and Technology, [3] Pompeu Fabra University
[4] HSE University, [5] Artificial Intelligence Research Institute
nikolay.babakov@usc.es

Abstract

This paper presents a solution to the GenChal 2022 shared task dedicated to feedback comment generation for writing learning. In terms of this task given a text with an error and a span of the error, a system generates an explanatory note that helps the writer (language learner) to improve their writing skills. Our solution is based on fine-tuning the T5 model on the initial dataset augmented according to syntactical dependencies of the words located within indicated error span. The solution of our team "nigula" obtained second place according to manual evaluation by the organizers.

1 Introduction

Feedback comment generation for language learners is the task of generating an explanatory note that helps the writers (language learners) improve their writing skills for text with error and a span of this error. In terms of GenChall2022[1], the target language is English, but this task applies to any language. Moreover, this task is mostly designed not for such cases as simple typos or misspellings (something which can be easily detected by standard grammar-error detection systems) but for erroneous, unnatural, or problematic words. See Table 1 for examples of such types of errors and corresponding comments.

Detecting the aforementioned misuse of words is not enough to prevent the same error in the future. It is important to provide some explanation and reference to the error-specific rule and give a direct hint on which correct word should be used in this particular case.

While any language follows certain rules which can be encoded manually to obtain the solution relying on rules and templates, in our work we try to use the benefit of the availability of parallel data and explore the limits of modern seq2seq models with the minimal number of rules and manual labor involved.

The contributions of our work are as follows:

- We present our solution based on tuning the T5-large model on the dataset augmented in the special error syntax-based approach.
- We opensource the model on Huggingface Model Hub[2] and code for experiments on GitHub[3].

2 Related works

The first attempts to provide feedback on a particular error type were based on rules (Nagata et al., 2014). In Morgado da Costa et al. (2020) authors used the English Resource Grammar parser to analyze the learner's sentence. If the parser fails to process a sentence, this sentence is supposed to have an error, and, so-called, mal-rules are used to detect the particular type of error. If the mal-rule works, the user is provided with a mal-rule-specific comment.

The steps towards the usage of more modern approaches, such as neural networks, have also been performed in this task. In Andersen et al. (2013), the authors use a combination of basic machine learning approaches to detect errors and rules to provide feedback on some common types of errors. Nagata (2019) showed that a neural retrieval-based method can be effective in preposition feedback comment generation. Lai and Chang (2019) proposed a method that provides context-sensitive explanations using grammatical error correction and templates. Gkatzia et al. (2014) suggested methods for automatically choosing feedback templates based on learning history. In Kaneko et al. (2022), the sentence with an error is first corrected with

*Work mostly has been done while at Skoltech
[1] https://fcg.sharedtask.org/task/
[2] https://huggingface.co/SkolkovoInstitute/GenChal_2022_nigula
[3] https://github.com/skoltech-nlp/feedback_generation_nigula

Learner's sentence	Golden feedback / Our system's output
Maybe at holiday and have free time then I can to have part-time job .	<Verbs> that follow an <auxiliary verb> are used in their <infinitive form> instead of a <to infinitive>.
	<verbs> that come right after an <auxiliary verb> are used in their <infinitive form>.
Become college student requires a lot of money .	A <verb phrase> needs to be converted into into a <noun phrase> in the form of a <to infinitive> or a <gerund> to be used as the <subject>.
	A <verb phrase> needs to be converted into into a <noun phrase> in the form of a <to infinitive> or a <gerund> to be used as the <subject>.
They might face with the danger of exploring of the battery or the problems of the electronic .	Since the <verb> «face» is a <transitive verb> and its <direct object> indicates the confronted object, it does not require a <preposition>.
	Since the <verb> «face» is a <transitive verb>, the <object> does not require a <preposition>.
There are many advantages to have a part-time job .	Use <preposition + gerund> instead of a <to-infinitive> to describe the "advantage". Look up the use of the <noun> «advantage» in a dictionary.
	Use the structure <preposition+gerund> instead of a <to-infinitive> with the <noun> «advantage».

Table 1: **Input and output (golden and system).** Example of feedback comments from the competition test set and generated without system. The word with an error is highlighted in red. < > signs indicate grammar terms, « » mean reference to the word in the learner text.

the grammar error correction system and then the K-nearest neighbors algorithm is used to provide the learner with the pair of an incorrect and corrected sentence which contains a similar kind of error. In Getman (2021), the authors use unusual n-grams, out-of-vocabulary words, and several pretrained models to find an error in the learner's text. This system does not provide text feedback in natural language, but it generates a structured report of found errors in the text.

The useful subtask of the feedback comment generation is grammar error classification. The information on the particular type of error made in the text could be used either directly by creating a template comment to this error or by using the error type as an additional signal in training data. One example of such work is Bryant et al. (2017) which automatically extracts the edits between parallel original and corrected sentences using a linguistically-enhanced alignment algorithm. In this paper, a rule-based framework that relies solely on dataset-agnostic information such as lemma and part-of-speech is developed as well. Beyond this, the paper of Choshen et al. (2020) uses universal dependencies syntactic representation scheme.

The main limitation of using the most modern text-to-text models had been the non-availability of parallel datasets with errors and corresponding annotation. In Pilan et al. (2020), a unique dataset where feedback comments on linking words were annotated was released. The dataset used in GenChal 2022 was collected in Nagata (2019); Nagata et al. (2020) for the English language. The main types of errors in this dataset are misuse of prepositions and other writing items such as discourse and lexical choice.

3 Task description

In this section, we introduce the formal definition of the task and the dataset provided for it.

3.1 Task definition

The system is provided with the text that by default contains an error. Moreover, the exact error span is provided as well. The output of the system should be the text which provides explanatory feedback on the error. If the system fails to generate the feedback it is supposed to return the <NO_COMMENT> string.

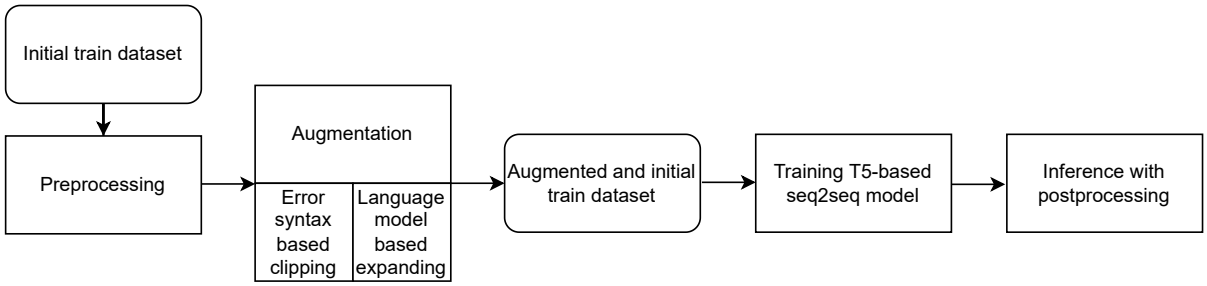

Figure 1: **Method workflow.** Description of the main steps of our feedback generation approach. The initial dataset is first preprocessed, then it is augmented by clipping the learner sentences according to the syntax relations of the word within error spans and the clipped sentences are then expanded with a large language model. The initial and augmented versions of the dataset are then used to train the T5 model in seq2seq mode. After that, the trained model is used to generate the feedback comments on the test data and the final text is post-processed.

3.2 Dataset

The form of the data in the task dataset is as follows:

- I agree it . \t 3:10 \t < <Agree > > is an <intransitive verb> and thus it requires a <preposition> before its object.

where \t stands for the tab character. If a sentence contains more than one error, it appears two or more times with different spans, so the input text always consists of only one sentence with one span. Also, the texts are pre-tokenized where tokens are separated by whitespace.

The feedback texts contain special symbols: <, > for grammatical terms (e.g. <intransitive verb>, <noun>, etc.) and < <, > > for citations of the words from learners' sentence (e.g. «agree»).

The dataset consists of the train (4867 samples) and development (169 samples) sets which were provided at the beginning of the competition and the test set (214 samples) which was provided in the last week of the competition and had only text and error span information.

4 Our method description

In this section, we introduce the main steps for training the model used for the final submission. These steps are also shown in Figure 1.

4.1 Preprocessing

Even though the text in the dataset is pre-tokenized, the special symbols described in 3.2 (<, >, < < , > >) can interfere with the tokenizer of a large pre-trained model. Thus, we lowercase the text and insert whitespace between words and the special symbols. Refer to Table 2 to have a look at the example of an initial and preprocessed sample.

As mentioned in 3.2, one sentence can have multiple errors, but according to the task definition, the system is supposed to provide feedback only to one slot of the error. To explicitly point out the exact error span in the learner's sentence we put similar special symbols (< < , > >) around the error span.

4.2 Augmentation

Even though the dataset has a limited amount of error types the variability of natural language yields an almost unlimited amount of situations in which each particular error can occur.

Let's demonstrate it in the following example of learner's text - *They can help their father or mother about money that we must use in the university too*. According to the span, the error is in the misuse of the preposition "about". To be more specific, the student has used an incorrect preposition after the "help + someone" construction. So, if we generate a new sentence that would be similar to the initial one by 'help someone about something' construction and would be different from other points of view, it will still correspond to the initial feedback and it could be applied to training the model in seq2seq mode as an additional training sample.

Our approach to augmentation consists of two parts. First, we cut the initial sentence by the last word which is syntactically related to the words within an error span. Second, we use the remaining text as a prompt to the language model, so it generates an alternative end to the sentence with a given prefix. Refer to Figure 2 for the principal scheme of the augmentation approach. More details can be found below in Sections 4.2.1 and 4.2.2.

Sentence	Comment
They can help their father or mother about money that we must use in the university too .	< <About > >is not the appropriate <preposition>to be used when a <noun>follows the structure <help + someone >. Look up the use of the <verb> < <help > > in a dictionary to learn the appropriate <preposition> to be used.
they can help their father or mother < < about > > money that we must use in the university too .	< < about > > is not the appropriate < preposition > to be used when a < noun > follows the structure < help + someone > . look up the use of the < verb > < < help > > in a dictionary to learn the appropriate < preposition > to be used .

Table 2: **Preprocessing.** Example of the preprocessing of learner's text and the corresponding feedback comment (the initial one is at the top, the preprocessed one is at the bottom).

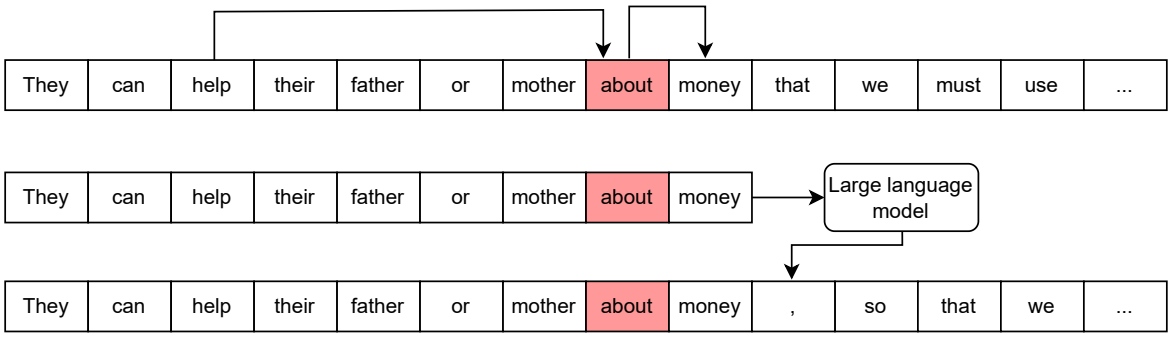

Figure 2: **Augmentation workflow.** The sentence with an error passes automatic syntactical analysis. The sentence is cropped by the last word that is syntactically connected to any word within the error span or by the error span itself (if no syntactically dependent words are located after the span). The cropped sentence is used as a prompt for a large language model to generate an alternative phrase with a similar error.

4.2.1 Learner sentence clipping

We use the spacy[4] package to perform syntactic parsing of the learner text. In the case of the sentence from our example, the error word "about" is syntactically connected to the verb "help" and the noun "money". We assume that these words form the skeleton of the error in this particular sentence. Thus, we crop everything after the last connected word 'money' and the cropped sentence becomes *they can help their father or mother about money*. If the syntactically connected words are located before the error span we crop the sentence by the error span.

4.2.2 Text generation

The cropped version of the text can be used as a prompt to infer an entire sentence with a large pre-trained language model. We use EleutherAI/gpt-neo-1.3B (Gao et al., 2020; Black et al., 2021)[5] to generate new texts.

This approach allows us to get a new sample consisting of the new sentence (which consists of the prefix similar to the initial one and the extension generated by the language model) and the feedback similar to the initial sample.

Here are some examples of the sentences generated by the model from the prompt *they can help their father or mother about money*:

- they can help their father or mother about

[4] https://spacy.io/models/en#en_core_web_md

[5] https://huggingface.co/EleutherAI/gpt-neo-1.3B

Exp. #	Data	BLEU
1	Initial	0.64
2	Augmented	0.65
3	Initial&augmented	0.67

Table 3: **Training steps.** Sequence of experiments conducted to train the final model. BLEU is shown for the validation dataset.

money, so that we can be independent. We have to work hard to earn our bread.

- they can help their father or mother about money." "Well, if I do that I'll have to buy clothes, and pay my own expense." "The girl has
- they can help their father or mother about money, they can help the mother, who can give us some medicine? We are able to keep the household alive from the old and the sick with

Some examples in the original dataset have similar feedback comments. If the learner's sentence is related to the group of samples which have ten or more similar feedback sentences, we assume that there is enough information for training the final model and do not apply augmentation to it.

Thus, we apply the augmentation technique to more than 4700 samples from the initial dataset. Each sample is augmented with 8-10 sentences. The final versions of the augmented dataset consist of 43,174 samples. We use these samples as additional data for training the final model.

4.3 Model training

The final solution is based on tuning T5 model (Raffel et al., 2020)[6]. The model's training input is the preprocessed sentences (see Section 4.1) and its target output is the corresponding feedback comments. Our default training parameters are batch size 8, Adam optimizer, gradient clipping by 1, and learning rate 1e-5. During training, we evaluate the current version of the model by calculating the BLEU score (Papineni et al., 2002) on the validation dataset.

As shown in Table 3 the main training steps are as follows:

- We train the model on the initial dataset. The best validation BLEU score is 0.64
- We tune the best version of the model on the augmented dataset (just newly generated samples). The best validation BLEU score is 0.66
- We merge both datasets, decrease the learning rate to 1e-6 and tune it for 4000 steps. This increases the validation BLEU score up to 0.67

The final version of the model is open-sourced to HuggingFace ModelHub.

4.4 Postprocessing

During inference, the final model demonstrates unusual behavior in terms of placing the special marks. To be more specific, the model correctly generates the closing brackets (>, > >) but fails to generate opening brackets (<, < <). We apply some postprocessing to the generated text. For doing this, we collect all possible grammar term references within train and validation texts, find the longest possible term before the closing bracket, and put opening brackets before the found term.

4.5 Other experiments

Except for the data augmentation approach, we also tried other hypotheses, which did not work well according to the preliminary analysis. We do not provide a full comparison with our main solution, but we find it useful to share them because this is the first generation challenge of feedback comment generation and there seems to be no or very little relevant work.

First, we tried various ways to enrich the training samples with auxiliary information, such as part of speech of the words inside the error span, corrected words, or grammar error classes (obtained with ERRANT[7] classification model). The results of this group of approaches were slightly worse than the main solution, however, we may assume that we did not dedicate much time to that, so it could be promising to conduct further experiments in this direction.

Second, we tried to find some easy heuristics that can be used for retrieving the existing suitable comment from the train data. We clustered learners' texts using similar words or parts of speech within the error span or vector representation of the text and then manually analyzed the feedback comments corresponding to every cluster. We also

[6] https://huggingface.co/t5-large

[7] https://pypi.org/project/errant/

Preprocessed input	Reference comments	System output	Manual evaluation by orgs
most < < of > > restaurant separate smoking are and non smoking are , but the smoking smell is remain this place .	Something defined follows < < most of > > . When referring to restaurants in general, use < < most > > as an <adjective>instead of a <noun>.	A group of something specific follows < < most of > > . Use < < most > > as an <adjective>rather than a <noun>when simply referring to something in general.	correct
most of restaurant separate smoking are and non smoking are , but the smoking smell is < < remain > > < < this > > place .	The <verb> < < remain > > is an <intransitive verb>and cannot take <direct objects>without a <preposition>of place.	When the <verb> < < remain > > is used as an <intransitive verb>, a <preposition>needs to precede the <object>. Look up the <verb> < < remain > > in the dictionary to find the appropriate <preposition>.	correct
part time job is a very important < < of > > people in present .	Use the <preposition>'for' to express "to be important for people".	It is not grammatically incorrect to use the <preposition> < < of > > , but using < < of > > emphasizes the part in one's opinion". It is more natural to use the <preposition>for'to express to be important for...".	incorrect
part time job is a very important of people < < in > > present .	The <preposition>used in the <idiom>with the <noun> < < present > > to express "now" is not < < in > > .	The <preposition>used in the <idiom>with the <noun> < < present > > to express for the moment" or now" is not < < in > > .	correct

Table 4: Examples of system output to similar sentences with different error span

tried to do similar experiments in the opposite direction (clustered feedback comments and analyze the learners' sentences). This approach let us find some heuristics that were used as an auxiliary tool for the language model-based feedback generation. However, the decision that used this tool with the trained model did not show any significant improvement over the pure model-based approach, which most probably means that such heuristics can be learned by the language model itself.

5 Results

The results of the system output are scored automatically and manually.

Automatic and manual scoring compares the system's outputs with manually created feedback comments. The automatic approach uses the BLEU score. In terms of manual scoring, a system output is regarded as appropriate if it contains information similar to the reference and does not contain information that is irrelevant to the offset; it may

contain information that the reference does not contain as long as it is relevant to the offset. If these conditions are met, the output is labeled as correct. The task definition (see Section 3.1) allows systems to generate <NO_COMMENT> phrase which is excluded from both the numerator and the denominator of precision and the numerator of recall. That is why the final score is calculated as precision, recall, and F1-score.

We do not make any filtering of the generated feedback, thus there is no case when our system generates <NO_COMMENT> phrase, so all metrics are equal. Refer to Table 5 for the results of manual evaluation by organizers of top-3 solutions. Our solution took second place.

It is also worth mentioning that the approach of using double brackets as a signal of the exact location of the error span to train the model worked well. To be more precise, the system always generates different feedback for similar sentences with different error spans and 12 out of 20 sentences in the test set (similar sentences with different slots were presented by pairs, so in total there are ten pairs of such samples) were scored as correct by organizers. Examples of such sentences can be found at Table 4.

6 Future work

There are several possible ways to improve the proposed data augmentation approach, that we leave for future work.

In our approach, we use the large language model to generate a new text using the prompt that contains a grammar error. The error could generally affect the quality of the generated text, which is why it could be interesting to first automatically correct the error in the clipped sentence, use the language model to generate a new version of the sentence, and then replace the corrected word with the erroneous word in the new sentence.

Another promising direction of the improvement of the augmentation approach is to apply changes not only to the right part of the error span but also to the left part. This could be done, for example by filling the masks placed on the position of random words that are not syntactically related to the words within the error span.

The amount of data generated for our experiments was based on a "the good the better" basis. However, it is also worth studying the relation between the amount of augmented data and the improvement in the quality of the trained model.

#	Team ID	Precision	Recall	F1
1	ihmana	0.6244	0.6186	0.6215
2	nigula (ours)	0.6093	0.6093	0.6093
3	TMUUED	0.6132	0.6047	0.6089

Table 5: **Results.** Manual evaluation by organizers.

7 Conclusion

In this paper, we present our solution for GenChal 2022 shared task dedicated to feedback comments generation to improve the English language learning experience. Our solution uses the error span based preprocessing of the learner's text, augmentation of the dataset by clipping of the learner's text w.r.t syntactic dependency to the words within the error span, and then the inference of large language model, using clipped text as a prompt, and finally training large T5-based model with both initial and augmented version of the dataset. Our solution took second place in this competition according to manual evaluation by organizers. The model and code of our experiments are open-sourced.

We also share the track of unsuccessful experiments and general ideas about alternative approaches to this task to prepare the ground for future researchers.

8 Acknowledements

This work was supported by a joint MTS-Skoltech laboratory on AI, the European Union's Horizon 2020 research and innovation program under the Marie Skodowska-Curie grant agreement No 860621, and the Galician Ministry of Culture, Education, Professional Training, and University and the European Regional Development Fund (ERDF/FEDER program) under grant ED431G2019/04.

References

Øistein E. Andersen, Helen Yannakoudakis, Fiona Barker, and Tim Parish. 2013. Developing and testing a self-assessment and tutoring system. In *Proceedings of the Eighth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 32–41, Atlanta, Georgia. Association for Computational Linguistics.

Sid Black, Gao Leo, Phil Wang, Connor Leahy, and Stella Biderman. 2021. GPT-Neo: Large

Scale Autoregressive Language Modeling with Mesh-Tensorflow. If you use this software, please cite it using these metadata.

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.

Leshem Choshen, Dmitry Nikolaev, Yevgeni Berzak, and Omri Abend. 2020. Classifying syntactic errors in learner language. In *Proceedings of the 24th Conference on Computational Natural Language Learning*, pages 97–107, Online. Association for Computational Linguistics.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, et al. 2020. The pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*.

Yaroslav Getman. 2021. Automated writing support for swedish learners. In *Swedish Language Technology Conference and NLP4CALL*, pages 21–26.

Dimitra Gkatzia, Helen Hastie, and Oliver Lemon. 2014. Comparing multi-label classification with reinforcement learning for summarisation of time-series data. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1231–1240, Baltimore, Maryland. Association for Computational Linguistics.

Masahiro Kaneko, Sho Takase, Ayana Niwa, and Naoaki Okazaki. 2022. Interpretability for language learners using example-based grammatical error correction.

Yi-Huei Lai and Jason Chang. 2019. TellMeWhy: Learning to explain corrective feedback for second language learners. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP): System Demonstrations*, pages 235–240, Hong Kong, China. Association for Computational Linguistics.

Luís Morgado da Costa, Roger V P Winder, Shu Yun Li, Benedict Christopher Lin Tzer Liang, Joseph Mackinnon, and Francis Bond. 2020. Automated writing support using deep linguistic parsers. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 369–377, Marseille, France. European Language Resources Association.

Ryo Nagata. 2019. Toward a task of feedback comment generation for writing learning. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3206–3215, Hong Kong, China. Association for Computational Linguistics.

Ryo Nagata, Kentaro Inui, and Shin'ichiro Ishikawa. 2020. Creating corpora for research in feedback comment generation. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 340–345, Marseille, France. European Language Resources Association.

Ryo Nagata, Mikko Vilenius, and Edward Whittaker. 2014. Correcting preposition errors in learner English using error case frames and feedback messages. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 754–764, Baltimore, Maryland. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Ildiko Pilan, John Lee, Chak Yan Yeung, and Jonathan Webster. 2020. A dataset for investigating the impact of feedback on student revision outcome. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 332–339, Marseille, France. European Language Resources Association.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.