Text Generation with Text-Editing Models

Eric Malmi[◇], **Yue Dong**[♣], **Jonathan Mallinson**[◇], **Aleksandr Chuklin**[◇],
Jakub Adamek[◇], **Daniil Mirylenka**[◇], **Felix Stahlberg**[◇],
Sebastian Krause[◇], **Shankar Kumar**[◇], **Aliaksei Severyn**[◇]
[◇]Google
[♣]McGill University & Mila
text-editing-tutorial@google.com

Abstract

Text-editing models have recently become a prominent alternative to seq2seq models for monolingual text-generation tasks such as grammatical error correction, simplification, and style transfer. These tasks share a common trait – they exhibit a large amount of textual overlap between the source and target texts. Text-editing models take advantage of this observation and learn to generate the output by predicting edit operations applied to the source sequence. In contrast, seq2seq models generate outputs word-by-word from scratch thus making them slow at inference time. Text-editing models provide several benefits over seq2seq models including faster inference speed, higher sample efficiency, and better control and interpretability of the outputs. This tutorial[1] provides a comprehensive overview of text-editing models and current state-of-the-art approaches, and analyzes their pros and cons. We discuss challenges related to productionization and how these models can be used to mitigate hallucination and bias, both pressing challenges in the field of text generation.

1 Introduction

After revolutionizing the field of machine translation (Sutskever et al., 2014; Cho et al., 2014; Bahdanau et al., 2015), sequence-to-sequence (seq2seq) methods have quickly become the standard approach for not only multilingual but also for *monolingual* sequence transduction / text generation tasks, such as text summarization, style transfer, and grammatical error correction. While delivering significant quality gains, these models, however, are prone to hallucinations (Maynez et al., 2020; Pagnoni et al., 2021). The seq2seq task setup (where targets are generated from scratch word by word) overlooks the fact that in many monolingual tasks the source and target sequences have a considerable overlap, hence targets could be reconstructed from the source inputs by applying a set of edit operations.

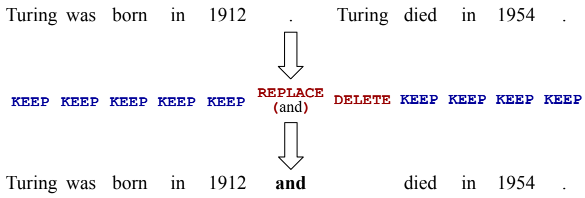

Figure 1: An example of using a text-editing approach to solve a sentence-fusion task.

Text-editing models attempt to address some of the limitations of seq2seq approaches and there has been recently a surge of interest in applying them to a variety of monolingual tasks including text simplification (Dong et al., 2019; Mallinson et al., 2020; Agrawal et al., 2021), grammatical error correction (Awasthi et al., 2019; Omelianchuk et al., 2020; Malmi et al., 2019; Stahlberg and Kumar, 2020; Rothe et al., 2021; Chen et al., 2020; Hinson et al., 2020; Gao et al., 2021), sentence fusion (Malmi et al., 2019; Mallinson et al., 2020) (see an example in Figure 1), MT automatic post-editing (Gu et al., 2019; Zietkiewicz, 2020; Mallinson et al., 2020), text style transfer (Reid and Zhong, 2021; Malmi et al., 2020), data-to-text generation (Kasner and Dušek, 2020), and utterance rewriting (Liu et al., 2020; Voskarides et al., 2020; Jin et al., 2022).

Text-editing approaches claim to be more accurate or on-par with seq2seq baselines especially in low resource settings, less prone to hallucinations and faster at inference time. These advantages have generated a substantial and continued level of interest in text-editing research. The goal of this tutorial is to provide the first comprehensive overview of the family of text-editing approaches and to offer practical guidelines for applying them to a variety of text-generation tasks.

[1]Website: https://text-editing.github.io/

Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics Human Language Technologies: Tutorial Abstracts, pages 1 - 7
July 10-15, 2022

Section	Duration
Introduction	15 min
What are text-editing models?	
Text-editing vs. seq2seq models	
Model design	40 min
Example model + model landscape	
Edit-operation types	
Tagging architecture	
Auto-regressiveness	
Converting target texts to target edits	
Applications	45 min
Overview	
Grammatical Error Correction	
Text Simplification	
Unsupervised Style Transfer	
Incomplete Utterance Rewriting	
Controllable generation	25 min
Mitigating hallucinations	
Controllable dataset generation	
Multilingual text editing	25 min
Tokenization	
Handling morphology	
Practical aspects	
Productionization	25 min
Latency	
Sample efficiency	
Recommendations and future directions	5 min
Total	180 min

Table 1: Tutorial structure and duration of each section.

1.1 Target Audience and Prerequisites

The tutorial is intended for researchers and practitioners who are familiar with generic seq2seq text-generation methods, such as Transformer (Vaswani et al., 2017) and pre-trained language models like BERT (Devlin et al., 2019). However, prior experience with text-editing models is not required to be able to follow the tutorial.

We expect the topic to attract people in both academia and industry. The high-sample efficiency and low-computational requirements of text-editing models (Malmi et al., 2019; Mallinson et al., 2020) makes them an attractive baseline, e.g., for researchers developing new text-generation tasks for which large training sets do not yet exist. Moreover, the high-inference speed of text-editing methods, owing to their often non-autoregressive architecture (Awasthi et al., 2019; Mallinson et al., 2020), makes them suitable for building real-time applications.

2 Tutorial Outline

The structure of the tutorial with duration estimates for different sections are shown in Table 1. Below we provide brief descriptions for each section.

Introduction. We first define the family of text-editing methods: Text-editing models are sequence-transduction methods that produce the output text by predicting edit operations which are applied to the inputs. In contrast, the traditional seq2seq methods produce the output from scratch, token by token. We summarize the main pros and cons of these two approaches and provide guidelines for choosing which approach is more suitable for a given task.

Model Design. The similarities and differences of a set of popular text-editing methods will be analyzed in terms of the types of edit operations they employ, their tagging architecture, and whether they are auto-regressive or feedforward. We also discuss methods for converting target texts into target edit sequences, a task which often does not have a unique solution. Table 2 provides a summary of the similarities and differences between the methods covered in the tutorial.

Applications. A key criterion for determining whether text-editing models are a good fit for a given application is the average degree of overlap between source and target texts. The higher the overlap, the more input tokens can be reused to generate the target, thus resulting in a simpler edit sequence. We give an overview of applications with a high degree of overlap to which text-editing methods have been applied to. Then we do a deep dive in to the following applications: grammatical error correction, text simplification, unsupervised style transfer, and incomplete utterance rewriting.

Controllable Generation. Text-editing models with a restricted vocabulary of phrases to insert (Malmi et al., 2019; Jin et al., 2022) or with linguistically informed suffix-transformation operations (Awasthi et al., 2019; Omelianchuk et al., 2020) are less prone to different types of hallucination since the models cannot produce arbitrary outputs. Moreover, the restricted vocabulary makes it feasible to manually refine the list of phrases that the model can insert. Another route through which the decomposition of the generation task into explicit edit operations can improve controllability is via biasing of certain types of edits to control how often the model will insert new text (Dong et al., 2019; Omelianchuk et al., 2020). Controllable generation with editing models can be useful for generating large synthetic datasets with a desired distribution of errors, which yields improvements in tasks such

Method	Non-autoregressive	Pre-trained decoder	Reordering	Unsupervised	Language-agnostic	Application(s)
EdiT5 (Mallinson et al., 2022)	(✓)	✓	✓		✓	*multiple*
EditNTS (Dong et al., 2019)					✓	Simplification
Felix (Mallinson et al., 2020)	✓	✓	✓		✓	*multiple*
GECToR (Omelianchuk et al., 2020)	✓	(✓)				GEC
HCT (Jin et al., 2022)	✓		✓		✓	Utterance Rewriting
LaserTagger (Malmi et al., 2019)	✓				✓	*multiple*
LevT (Gu et al., 2019)	(✓)	✓			✓	*multiple*
LEWIS (Reid and Zhong, 2021)		✓		✓	✓	Style Transfer
Masker (Malmi et al., 2020)	✓	✓		✓	✓	*multiple*
PIE (Awasthi et al., 2019)	✓	✓				GEC
Seq2Edits (Stahlberg and Kumar, 2020)					(✓)	*multiple*
SL (Alva-Manchego et al., 2017)	✓		✓		✓	Simplification

Table 2: Overview of selected text-editing methods.

as grammatical error correction (Stahlberg and Kumar, 2021). We will provide concrete examples of the aforementioned control measures and their effects.

Multilingual Text Editing. Most text-editing models, like text-generation models in general, are evaluated on English, but there are also methods evaluated or specifically developed for other languages, including Chinese (Hinson et al., 2020; Liu et al., 2020), Czech (Náplava and Straka, 2019), German (Mallinson et al., 2020), Russian (Stahlberg and Kumar, 2020), and Ukrainian (Syvokon and Nahorna, 2021). Apart from general tokenization-related challenges discussed in (Mielke et al., 2021), an additional challenge with applying text-editing methods to morphologically rich languages is a potential mismatch between the subword tokens, on which the underlying sequence labeling model operates, and the morphemes or affixes, on which the edits should happen. Possible solutions to this challenge include developing custom inflection operations (Awasthi et al., 2019; Omelianchuk et al., 2020) or learning them from the data (Straka et al., 2021), and using more fine-grained edit operations, such as character-level edits (Gao et al., 2021).

An additional challenge when building a truly multilingual model—as opposed to one model per language—is to ensure that it is not skewed towards a particular language or a set of languages (Chung et al., 2020) while being computationally efficient.

Productionization. We discuss how casting a text-generation problem as a text-editing task often allows the use of significantly faster and more data-efficient model architectures, without sacrificing output quality. We make use of the TensorFlow Profiler[2] to compare latencies of text-editing and non-text-editing solutions for an example problem, and illustrate where the time savings come from.

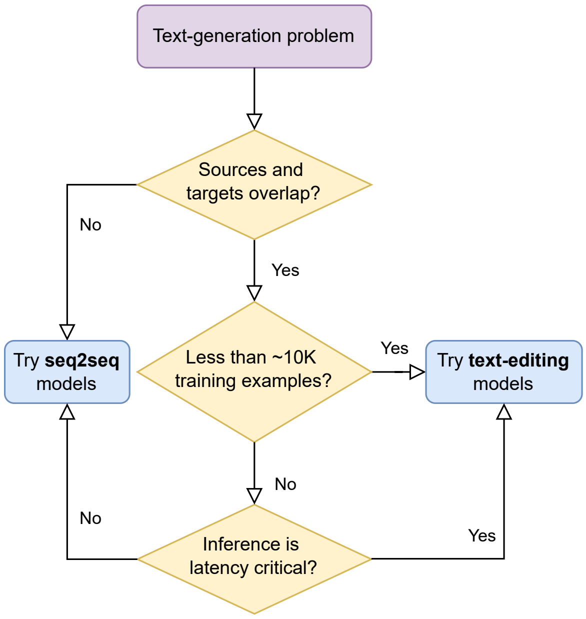

Figure 2: Proposed flowchart for deciding when to try a text-editing approach.

Recommendations and Future Directions. We provide practical guidelines for when to use (and when not to use) text-editing methods (see Figure 2 for a summary). We also outline possible future directions which include: (i) learned edit operations, (ii) studying the effects of different subword segmentation methods since these typically determine the granularity at which the edit operations are applied, (iii) text-editing-specific pre-training methods, (iv) sampling strategies for text-editing methods, and (v) studying the effects of scaling up

[2] https://www.tensorflow.org/guide/profiler#trace_viewer_interface

text-editing methods, a strategy that has been found to be very effective for many other text-generation methods (Brown et al., 2020; Chowdhery et al., 2022).

3 Diversity Considerations

A significant portion of the tutorial is devoted to discussing multilingual text-editing, including applying text-editing models to morphologically rich languages which presents specific challenges related to larger vocabularies and the need to edit word affixes. The presenters come from both academia and industry, are native speakers of 8 languages based in 4 different countries (Switzerland, Germany, Canada, USA), and are of different seniority levels from a PhD student to a Senior Staff Research Scientist.

4 Reading List

Before the tutorial, we expect the audience to read (Vaswani et al., 2017) and (Devlin et al., 2019). For references to text-editing works that will be discussed in the tutorial, see Table 2.

Breadth. 50% of the methods that will be discussed in the tutorial (cf. Table 2) are developed by different subsets of the tutorial instructors.

5 Presenters

Eric Malmi is a Senior Research Scientist at Google Switzerland. His research is focused on developing text-generation models for grammatical error correction and text style transfer. He received his PhD from Aalto University, Finland, where he also taught a course on Recent Advances in Natural Language Generation in Spring 2022.

Yue Dong is a final-year PhD student in CS at McGill University and Mila, Canada. Her research is focused on conditional text generation. She is a co-organizer for the NewSum workshop at EMNLP 2021 and ENLSP workshop at NeurIPS 2021.

Jonathan Mallinson is a Research Engineer at Google Switzerland. His research is focused on low-latency text-to-text generation. He received his PhD from the University of Edinburgh, Scotland.

Aleksandr Chuklin is a Research Engineer at Google Switzerland. His current research focuses on multi-lingual NLG. He organized workshops and conducted tutorials at conferences such as SIGIR, EMNLP, and IJCAI. Aleksandr received his PhD from University of Amsterdam, The Netherlands.

Jakub Adamek is a Research Engineer at Google Switzerland focusing on grammatical error correction and low-latency models. He received his MSc from Jagiellonian University.

Daniil Mirylenka is a Research Engineer at Google Switzerland working on text editing with application to grammatical error correction. He received his PhD from the University of Trento, Italy.

Felix Stahlberg is a Research Scientist at Google focusing on grammatical error correction and text style models. He received his PhD from Cambridge University, UK.

Sebastian Krause is a Senior Research Engineer at Google Switzerland. His work is focused on multi-lingual rewriting of questions in low-latency settings. Sebastian received his PhD in Engineering from the Technical University of Berlin, Germany.

Shankar Kumar is a Senior Staff Research Scientist at Google leading a research team working on speech and language algorithms. He received his PhD from the Johns Hopkins University, US.

Aliaksei Severyn is a Staff Research Scientist at Google Switzerland leading an applied research team working on next generation NLG solutions. He received his PhD from University of Trento, Italy.

6 Ethical Considerations

Text-generation methods have the potential to generate non-factual (Maynez et al., 2020; Pagnoni et al., 2021; Kreps et al., 2020) and offensive content (Gehman et al., 2020). Furthermore, training these models on uncurated data can lead to the models replicating harmful views presented in the training data (Bender et al., 2021). Text-editing models are also susceptible to these issues, but they have been shown to mitigate some of them. Specifically, they reduce the likelihood of different types of hallucination (Malmi et al., 2019) and their higher sample efficiency (Malmi et al., 2019; Mallinson et al., 2020) enables more careful curation of the training data. The tutorial will discuss the ethical issues related to text generation and provide concrete examples on how text-editing models can help mitigate them.

References

Sweta Agrawal, Weijia Xu, and Marine Carpuat. 2021. A non-autoregressive edit-based approach to controllable text simplification. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3757–3769, Online. Association for Computational Linguistics.

Fernando Alva-Manchego, Joachim Bingel, Gustavo Paetzold, Carolina Scarton, and Lucia Specia. 2017. Learning how to simplify from explicit labeling of complex-simplified text pairs. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 295–305, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Abhijeet Awasthi, Sunita Sarawagi, Rasna Goyal, Sabyasachi Ghosh, and Vihari Piratla. 2019. Parallel iterative edit models for local sequence transduction. 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 4260–4270, Hong Kong, China. Association for Computational Linguistics.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Emily M Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, pages 610–623.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Mengyun Chen, Tao Ge, Xingxing Zhang, Furu Wei, and Ming Zhou. 2020. Improving the efficiency of grammatical error correction with erroneous span detection and correction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7162–7169, Online. Association for Computational Linguistics.

Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar. Association for Computational Linguistics.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2022. PaLM: Scaling Language Modeling with Pathways. *arXiv preprint arXiv:2204.02311*.

Hyung Won Chung, Dan Garrette, Kiat Chuan Tan, and Jason Riesa. 2020. Improving multilingual models with language-clustered vocabularies. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4536–4546, Online. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Yue Dong, Zichao Li, Mehdi Rezagholizadeh, and Jackie Chi Kit Cheung. 2019. EditNTS: An neural programmer-interpreter model for sentence simplification through explicit editing. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3393–3402, Florence, Italy. Association for Computational Linguistics.

Mengyi Gao, Canran Xu, and Peng Shi. 2021. Hierarchical character tagger for short text spelling error correction. In *Proceedings of the Seventh Workshop on Noisy User-generated Text (W-NUT 2021)*, pages 106–113.

Samuel Gehman, Suchin Gururangan, Maarten Sap, Yejin Choi, and Noah A. Smith. 2020. RealToxicityPrompts: Evaluating neural toxic degeneration in language models. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3356–3369, Online. Association for Computational Linguistics.

Jiatao Gu, Changhan Wang, and Junbo Zhao. 2019. Levenshtein transformer. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 11179–11189.

Charles Hinson, Hen-Hsen Huang, and Hsin-Hsi Chen. 2020. Heterogeneous recycle generation for Chinese grammatical error correction. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2191–2201, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Lisa Jin, Linfeng Song, Lifeng Jin, Dong Yu, and Daniel Gildea. 2022. Hierarchical context tagging for utterance rewriting. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022*. AAAI Press.

Zdeněk Kasner and Ondřej Dušek. 2020. Data-to-text generation with iterative text editing. In *Proceedings of the 13th International Conference on Natural Language Generation*, pages 60–67, Dublin, Ireland. Association for Computational Linguistics.

Sarah Kreps, R Miles McCain, and Miles Brundage. 2020. All the news that's fit to fabricate: Ai-generated text as a tool of media misinformation. *Journal of Experimental Political Science*, pages 1–14.

Qian Liu, Bei Chen, Jian-Guang Lou, Bin Zhou, and Dongmei Zhang. 2020. Incomplete utterance rewriting as semantic segmentation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2846–2857, Online. Association for Computational Linguistics.

Jonathan Mallinson, Jakub Adamek, Eric Malmi, and Aliaksei Severyn. 2022. Edit5: Semi-autoregressive text-editing with t5 warm-start. *arXiv preprint arXiv:2205.12209*.

Jonathan Mallinson, Aliaksei Severyn, Eric Malmi, and Guillermo Garrido. 2020. FELIX: Flexible text editing through tagging and insertion. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1244–1255, Online. Association for Computational Linguistics.

Eric Malmi, Sebastian Krause, Sascha Rothe, Daniil Mirylenka, and Aliaksei Severyn. 2019. Encode, tag, realize: High-precision text editing. 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 5054–5065, Hong Kong, China. Association for Computational Linguistics.

Eric Malmi, Aliaksei Severyn, and Sascha Rothe. 2020. Unsupervised text style transfer with padded masked language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8671–8680, Online. Association for Computational Linguistics.

Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan McDonald. 2020. On faithfulness and factuality in abstractive summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1906–1919, Online. Association for Computational Linguistics.

Sabrina J Mielke, Zaid Alyafeai, Elizabeth Salesky, Colin Raffel, Manan Dey, Matthias Gallé, Arun Raja, Chenglei Si, Wilson Y Lee, Benoît Sagot, et al. 2021. Between words and characters: A brief history of open-vocabulary modeling and tokenization in nlp. *arXiv preprint arXiv:2112.10508*.

Jakub Náplava and Milan Straka. 2019. Grammatical error correction in low-resource scenarios. In *Proceedings of the 5th Workshop on Noisy User-generated Text (W-NUT 2019)*, pages 346–356, Hong Kong, China. Association for Computational Linguistics.

Kostiantyn Omelianchuk, Vitaliy Atrasevych, Artem Chernodub, and Oleksandr Skurzhanskyi. 2020. GECToR – grammatical error correction: Tag, not rewrite. In *Proceedings of the Fifteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 163–170, Seattle, WA, USA Online. Association for Computational Linguistics.

Artidoro Pagnoni, Vidhisha Balachandran, and Yulia Tsvetkov. 2021. Understanding factuality in abstractive summarization with FRANK: A benchmark for factuality metrics. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4812–4829, Online. Association for Computational Linguistics.

Machel Reid and Victor Zhong. 2021. LEWIS: Levenshtein editing for unsupervised text style transfer. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3932–3944, Online. Association for Computational Linguistics.

Sascha Rothe, Jonathan Mallinson, Eric Malmi, Sebastian Krause, and Aliaksei Severyn. 2021. A simple recipe for multilingual grammatical error correction. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 702–707, Online. Association for Computational Linguistics.

Felix Stahlberg and Shankar Kumar. 2020. Seq2Edits: Sequence transduction using span-level edit operations. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5147–5159, Online. Association for Computational Linguistics.

Felix Stahlberg and Shankar Kumar. 2021. Synthetic data generation for grammatical error correction with tagged corruption models. In *Proceedings of the 16th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 37–47, Online. Association for Computational Linguistics.

Milan Straka, Jakub Náplava, and Jana Straková. 2021. Character transformations for non-autoregressive gec tagging. In *Proceedings of the Seventh Workshop on Noisy User-generated Text (W-NUT 2021)*, pages 417–422.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 3104–3112.

Oleksiy Syvokon and Olena Nahorna. 2021. Ua-gec: Grammatical error correction and fluency corpus for the ukrainian language.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Nikos Voskarides, Dan Li, Pengjie Ren, Evangelos Kanoulas, and Maarten de Rijke. 2020. Query resolution for conversational search with limited supervision. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020*, pages 921–930. ACM.

Tomasz Zietkiewicz. 2020. Post-editing and rescoring of asr results with edit operations tagging. In *Proceedings of the PolEval2020 Workshop*.