# Predicting Compact Phrasal Rewrites with Large Language Models for Automatic Speech Recognition Post Editing

Anonymous ACL submission

## Abstract

Large Language Models (LLMs) excel at rewriting tasks such as text style transfer and grammatical error correction. Although the output in these tasks often significantly overlaps with the input, the decoding cost still increases with output length, regardless of the number of overlaps. By leveraging the overlap between the input and the output, Kaneko and Okazaki (2023) proposed model-agnostic edit span representations to compress the rewrites to save computation. They reported an output length reduction rate of nearly 80% with minimal accuracy impact in four rewriting tasks. In this paper, we propose alternative edit *phrase* representations inspired by phrase-based statistical machine translation. We systematically compare our phrasal representations with their span representation. We apply the LLM rewriting model to the task of Automatic Speech Recognition (ASR) post editing and show that our target-phrase-only edit representation has the best efficiency-accuracy trade-off. On the LibriSpeech test set, our method closes 50-60% of the WER gap between the edit span model and the full rewrite model while losing only 10-20% of the length reduction rate of the edit span model.

## 1 Introduction

Large Language Models pretrained on vast amount of texts and then fine-tuned, instruction-tuned, or prompted for generation tasks have achieved great success in the past few years (Raffel et al., 2020; Brown et al., 2020; Chowdhery et al., 2022; Anil et al., 2023; OpenAI et al., 2024). These models excel at text rewriting tasks, and text style transfer (Reif et al., 2022) and grammatical error correction (Rothe et al., 2021; Fang et al., 2023) in particular.

But the superior quality of these models comes along with a steep increase in the cost of computation. To enable broad deployment for a large user base, it is crucial to reduce the computational cost while maintaining the accuracy.

One of the common characteristics of the above-mentioned rewriting tasks is that their output often repeats spans of text in the input. Exploiting the common sub-strings between the input and the output can result in more compact representations for the output. LLMs can be fine-tuned on examples that map input to their compact rewrite representations instead of plain rewrites. At inference time, decoding output needs to be composed with the input to expand into complete rewrites. Kaneko and Okazaki (2023) gave one such representation, which is a numerical span indexing into the input sequence followed by a target phrase that will substitute the source phrase in the given span. We propose two new alternative representations. The first one uses a source-target phrase pair to represent each rewrite pattern, analogous to phrase pairs used by phrase-based statistical machine translation (Koehn et al., 2003). The second one only uses a target phrase along with left and right context words that appear in the input. We call the new representations *phrase* representations to distinguish them from the *span* representation of Kaneko and Okazaki (2023).

The clear advantage of compact representations over complete rewrite is that the number of decoding steps, and hence the computational cost of inference, is reduced. Kaneko and Okazaki (2023) reported an output length reduction rate of 80%. The disadvantage of such representations is that decoding errors can also lead to inconsistency with the input sequence in the expansion stage, causing an error propagation effect. For example, using the numerical span representation, if the left index or the right index is off by one, the source phrase to be substituted will also be off by one. The concatenation of the context and the substitution can therefore become disfluent. With phrase representations, context words are provided before and after substitution phrases, which can alleviate the problem of disfluency upon substitution. However, phrase

representations have their own problems too. The predicted context phrase may not match the input, which makes it necessary to discard the subsequent rewrite. The main focus of the paper is to evaluate the efficiency-accuracy trade-off of the different representations.

We choose Automatic Speech Recogintion (ASR) post editing as the task for applying LLM-based rewrite models and report word error rates (WER) and output length reduction rates using the span representation and the full rewrite models as the baselines.

Our contributions include the following.

- We propose a compact edit string representation with superior efficiency-accuracy trade-off than Kaneko and Okazaki (2023).
- We apply edit representation based rewriting LLMs to the task of ASR output correction. To the best of our knowledge, our work is the first to combine compact rewriting with generative LLMs to achieve substantial ASR WER reduction with manageable decoding cost.

## 2 Rewrite Representations

Mathematically speaking, for rewrite examples $(\mathbf{x}, \mathbf{y})$, where $\mathbf{x}$ is the input string and $\mathbf{y}$ is the output string, there is a compression function C and an expansion function E satisfying

$$\mathrm{E}(\mathbf{x}, \mathrm{C}(\mathbf{x}, \mathbf{y})) = \mathbf{y} \quad (1)$$

with the constraint that $|\mathrm{C}(\mathbf{x}, \mathbf{y})|$ has a much smaller average value than $|\mathbf{y}|$. At training time, examples are converted to $(\mathbf{x}, \hat{\mathbf{y}} = \mathrm{C}(\mathbf{x}, \mathbf{y}))$. At inference time, the final output is obtained by applying the expansion function $\mathrm{E}(\mathbf{x}, \mathbf{y}')$ where $\mathbf{y}'$ is the decoding output for $\mathbf{x}$. The edit representations in this section differ in the choice of the function pair C and E.

### 2.1 Edit Span Representation

The span representation of Kaneko and Okazaki (2023) is derived from a word alignment graph $\mathbf{a}$ between $\mathbf{x}$ and $\mathbf{y}$. Given a bipartite alignment graph between the input sequence and the output sequence, we can identify pairs of word spans between the two sides. Each span pair is a local rewrite instance indicating that the source span is substituted by the target span. In practice, the alignment is derived from the Levenshtein distance algorithm with the guarantee that the alignment links are monotonically ordered. It is always feasible to represent the entire rewrite as a sequence of local rewrite spans. For LLMs to predict the rewrites, we need a string representation of the span pairs. In their paper, the span representation is is specified as $(i, j, \mathbf{y}_{\mathbf{a}(i...j)})$, where $\mathbf{a}(i...j)$ is the corresponding target span of a source span $i...j$ and $\mathbf{y}_{\mathbf{a}(i...j)}$ is the target phrase in this span. Under this representation, C is the concatenation of the ordered span representations:

$$\mathrm{C} = \oplus_{(i,j)\in\mathbf{a}}(i, j, \mathbf{y}_{\mathbf{a}(i...j)}). \quad (2)$$

E is the program of applying the ordered local rewrites to the input sequence.

### 2.2 Phrase Pair Representation

The representation in Equation 2 is concise. It uses a pair of integers to represent a source span. However, this implies that LLMs have to count source tokens and generate indexing integer tokens interleaved with content tokens following the predefined format. The structured representation introduces brittleness to the model. A prediction error in the integer token sub-sequence can have a cascading effect when the entire output rewrite string is parsed and applied on the input. Instead, we resort to a natural language representation, which uses the source phrase $\mathbf{x}_{i...j}$ for a span $(i, j)$ directly as the prefix for the target phrase $\mathbf{y}_{\mathbf{a}(i...j)}$. However, one downside of our representation is that when the span $(i, j)$ is small, the subsequence $\mathbf{x}_{i...j}$ can be ambiguous, introducing errors into the expansion step. A solution is to add more context to $\mathbf{x}_{(i...j)}$ as well as $\mathbf{y}_{\mathbf{a}(i...j)}$ to make it much less likely to be ambiguous by extending the phrase pair to both the left and the right.

Formally, the new function C is

$$\mathrm{C} = \oplus_{(i,j)\in\mathbf{a}}(\mathtt{W}: \mathbf{x}_{(i-k...j+k)}, \mathbf{y}_{\mathbf{a}(i-k...j+k)}), \quad (3)$$

where $k$ is called the *dilation span*, W is a natural language prompt word like `rewrite`. The expansion function E involves parsing the pattern and prefix string matching and replacement on the input.

### 2.3 Target Phrase Representation

The representation in Equation 3 using both source phrases and target phrases has the disadvantage

| | |
|---|---|
| *source text* | Since we do not to bring cash to pay for the transportation fee , enormous time have been saved |
| *target text* | Since we do not *need* to bring cash to pay for the transportation fee , enormous time *has* been saved |
| *span* | 4 4 need, 16 17 has |
| *phrase pair* | rewrite: not to, not need to, rewrite: time have been, time has been |
| *target only* | rewrite: not need to, rewrite: time has been |

Table 1: Example output under various representations. We underline the numerical spans or substrings to be matched against the source text in the expansion stage.

of being verbose. Dilation spans on both sides are intended to make phrases less ambiguous can exacerbate the problem. However, dilation spans on target phrases can often be sufficient for disambiguation when the span size is three or higher. We can ignore source phrases and just use dilated target phrases as they contain both anchor text in the input and replacement text in the output. The following is the new compression function.

$$\mathrm{C} = \oplus_{(i,j)\in\mathbf{a}}(\mathtt{W}: \mathbf{y}_{\mathbf{a}(i-k\ldots j+k)}) \quad (4)$$

String matching and replacement in the implementation of function E deals with discontiguous dilation spans in the form of $\mathbf{y}_{\mathbf{a}(i-k\ldots i-1)}\cdots\mathbf{y}_{\mathbf{a}(j+1,j+k)}$.

In Table 1, we show actual examples of edit representations. Under the target only representation, "not ... to" has two matches ("not to" and "not to bring cash to"') in the source text. For such cases, we prefer the leftmost and closest pair to break ties.

## 3 Experiments

We use a decoder-only LLM for the task of correcting the output of a fast first pass Automatic Speech Recognition (ASR) model. The first pass model is a streaming model that decodes as audio comes in without the full context of the future. Therefore there is enough room for error correction using a pre-trained LLM with the full ASR transcription as the input. The task can be viewed as a variant of grammatical error correction (Brockett et al., 2006).

The ASR model we use is the Google USM model (Zhang et al., 2023). The LLMs we use are the PaLM 2 Gecko and Otter models (Anil et al., 2023). We fine-tune LLMs on the LibriSpeech (Panayotov et al., 2015) training set using the dev set for hyper-parameter and checkpoint selection and the test sets for final comparisons. The ASR model is frozen in our experiments. We fine-tune the entire Transformer LLM model to minimize

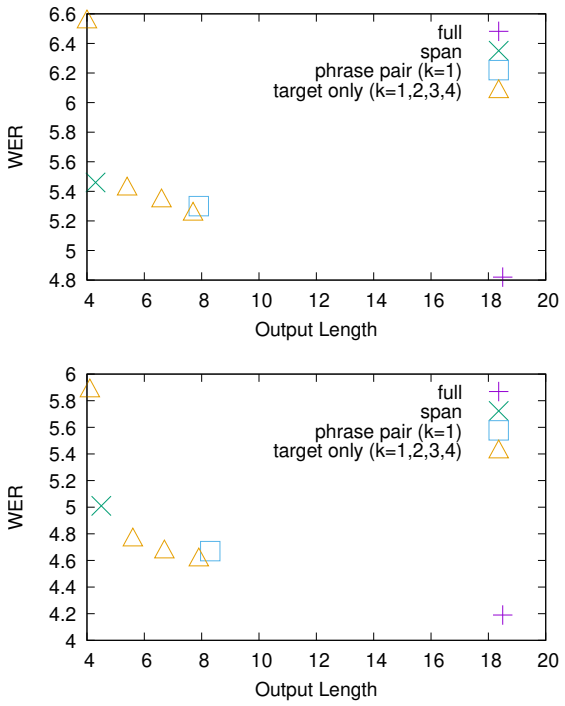


Figure 1: WER versus output length on the dev set. Top: PaLM 2 Gecko model. Bottom: PaLM 2 Otter model. *target only* with dilation span size 3 (the third △ from the left) is the best strategy.

the cross entropy loss on the transcription reference set given the ASR transcription generated by the frozen USM model as the prefix to the LLM decoder. We use two baselines. One is *full rewrite* model that uses the reference transcription directly. The other is *span rewrite* of Kaneko and Okazaki (2023) that uses the representation in Section 2.1. We are interested in two metrics. The quality metric is word error rate (WER) after expanding edit representations. The efficiency metric is decoder output length reduction rate.

Table 2 summarizes the main results. We show that the span representation indeed incurs more accuracy loss than the phrase representations. On the clean test set, *target only* is able to close 57% of the accuracy gap between *span* and *full*, while losing 12.5% of the length reduction rate. On the

| | test-clean *WER* | *Avg Output Length* | test-other *WER* | *Avg Output Length* |
|---|---|---|---|---|
| *USM* | 6.6 | - | 11.4 | |
| *full* | 2.7 (-59%) | 20 | 6.2 (-46%) | 18 |
| *span* | 3.4 (-48%) | 4 (-80%) | 7.5 (-34%) | 5 (-72%) |
| *phrase pair* | 3.0 (-54%) | 7 (-65%) | 7.1 (-38%) | 10 (-44%) |
| *target only* | 3.0 (-54%) | 6 (-70%) | 6.8 (-40%) | 8 (-56%) |

Table 2: Results of ASR (USM) post editing models based on PaLM 2 Otter. *full* has the lowest WER but has a high computational cost proportional to the average output length. *span* (Kaneko and Okazaki, 2023) is most efficient with the fewest output tokens. *target only* closes most of the WER gap between *span* and *full* while approaching the length reduction rate of *span*.

noisier other test set, *target only* is able to close 54% of the accuracy gap, while losing 22.2% of the length reduction rate.

### 3.1 Efficiency and Accuracy Trade-offs

In Figure 1, we plot WER versus output length for two model sizes: Gecko and Otter, and varying values of the phrase dilation hyper-parameter $k$ in Equation 3 and Equation 4. Both the phrase pair and the target phrase only strategies yield lower WER with slightly longer outputs than the span strategy. Overall, when $k$ is 3, the target phrase only strategy has the best trade-off. The trend stays across the two PaLM 2 model sizes.

### 3.2 Recovery Rate

In Section 2, we formulated the problem as selecting a pair of compression function C and expansion function E to satisfy Equation 1. The span representation is exact and unambiguous. So when E is applied, the equality is satisfied for all training examples. The phrase representations can be ambiguous and depend on the dilation spans to minimize the chance of multiple matches when the expansion function is applied. Table 3 summarizes the recovery rates, which is the percentage of examples in the dev set that satisify Equation 1. The phrase pair representation has sufficient source context in the source phrase so that its recovery rate is very close to 100%. For the target only representation, word bigrams ($k = 2$) or trigrams ($k = 3$) surrounding target phrases are sufficient for uniquely identifying their source side counterparts in most cases.

| *representation* | *recovery rate* |
|---|---|
| *phrase pair* ($k = 1$) | 99.98% |
| *target only* ($k = 1$) | 96.80% |
| *target only* ($k = 2$) | 99.50% |
| *target only* ($k = 3$) | 99.80% |
| *target only* ($k = 4$) | 99.80% |

Table 3: Recovery rate of phrase representations.

## 4 Related Work

Orthogonal efforts to speed up decoding include speculative decoding (Leviathan et al., 2023; Chen et al., 2023). They leverage the overlap in output distributions between a less accurate faster model and a more accurate slower model as well as hardware accelerators for parallel computing. They do not incur accuracy loss and are not limited to rewriting tasks. Combining compact representations with speculative decoding has the potential for even more speedups.

LLMs have been used for ASR correction in ranking and generation (Pu et al., 2023). Leng et al. (2021) model edit operations for efficient non-autoregressive decoding. It is possible to do hybrid decoding with LLMs: predicting which spans need to be rewritten followed by auto-regressive decoding of output rewrites.

## 5 Conclusions

We propose two edit phrase representations for rewriting tasks that compactly represent the differences between input and output strings. We use LLMs to predict such edits and expand the edits into complete rewrites with a deterministic string matching and replacement algorithm. Our work is a further development of the span representation by Kaneko and Okazaki (2023). For the task of ASR post editing, we close 50-60% of the WER gap between the most efficient model and the most accurate model, while only slowing down decoding by 10-20% relative to the efficient representation.

## Limitations

Concise edit representations presented in the paper are derived from the Levenshtein distance algorithm. The phrases are not linguistically meaningful or optimal from machine learning point of view. They are only minimal according to the edit distance. Going beyond edit distance to use differentiable functions for compression and expansion is an interesting open area for research.

The dilation spans we use to anchor phrases in the input are applied uniformly and equally on the left and right of each span of interest. It is likely that longer left context is more useful than right context since the decoder progresses from left to right.

We have not explored different formats for the rewrite phrases.

The expansion stage of the target only representation is more involved than the other two compact representations. Efficient data structures and string matching algorithms are necessary to take account of two discontiguous word spans.

Finally, we have not experimented with the latest and largest LLMs. It is possible that prompt engineering is sufficient to let these models generate concise rewrites. It is to be seen if the gap between full rewrite and edit representations can be reduced further with very large LLMs.

## References


Rohan Anil, Andrew M. Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, Eric Chu, Jonathan H. Clark, Laurent El Shafey, Yanping Huang, Kathy Meier-Hellstern, Gaurav Mishra, Erica Moreira, Mark Omernick, Kevin Robinson, Sebastian Ruder, Yi Tay, Kefan Xiao, Yuanzhong Xu, Yujing Zhang, Gustavo Hernandez Abrego, Junwhan Ahn, Jacob Austin, Paul Barham, Jan Botha, James Bradbury, Siddhartha Brahma, Kevin Brooks, Michele Catasta, Yong Cheng, Colin Cherry, Christopher A. Choquette-Choo, Aakanksha Chowdhery, Clément Crepy, Shachi Dave, Mostafa Dehghani, Sunipa Dev, Jacob Devlin, Mark Díaz, Nan Du, Ethan Dyer, Vlad Feinberg, Fangxiaoyu Feng, Vlad Fienber, Markus Freitag, Xavier Garcia, Sebastian Gehrmann, Lucas Gonzalez, Guy Gur-Ari, Steven Hand, Hadi Hashemi, Le Hou, Joshua Howland, Andrea Hu, Jeffrey Hui, Jeremy Hurwitz, Michael Isard, Abe Ittycheriah, Matthew Jagielski, Wenhao Jia, Kathleen Kenealy, Maxim Krikun, Sneha Kudugunta, Chang Lan, Katherine Lee, Benjamin Lee, Eric Li, Music Li, Wei Li, YaGuang Li, Jian Li, Hyeontaek Lim, Hanzhao Lin, Zhongtao Liu, Frederick Liu, Marcello Maggioni, Aroma Mahendru, Joshua Maynez, Vedant Misra, Maysam Moussalem, Zachary Nado, John Nham, Eric Ni, Andrew Nystrom, Alicia Parrish, Marie Pellat, Martin Polacek, Alex Polozov, Reiner Pope, Siyuan Qiao, Emily Reif, Bryan Richter, Parker Riley, Alex Castro Ros, Aurko Roy, Brennan Saeta, Rajkumar Samuel, Renee Shelby, Ambrose Slone, Daniel Smilkov, David R. So, Daniel Sohn, Simon Tokumine, Dasha Valter, Vijay Vasudevan, Kiran Vodrahalli, Xuezhi Wang, Pidong Wang, Zirui Wang, Tao Wang, John Wieting, Yuhuai Wu, Kelvin Xu, Yunhan Xu, Linting Xue, Pengcheng Yin, Jiahui Yu, Qiao Zhang, Steven Zheng, Ce Zheng, Weikang Zhou, Denny Zhou, Slav Petrov, and Yonghui Wu. 2023. Palm 2 technical report.

Chris Brockett, William B. Dolan, and Michael Gamon. 2006. Correcting ESL errors using phrasal SMT techniques. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 249–256, Sydney, Australia. Association for Computational Linguistics.

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.

Charlie Chen, Sebastian Borgeaud, Geoffrey Irving, Jean-Baptiste Lespiau, Laurent Sifre, and John Jumper. 2023. Accelerating large language model decoding with speculative sampling.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck,

Jeff Dean, Slav Petrov, and Noah Fiedel. 2022. Palm: Scaling language modeling with pathways.

Tao Fang, Shu Yang, Kaixin Lan, Derek F. Wong, Jinpeng Hu, Lidia S. Chao, and Yue Zhang. 2023. Is chatgpt a highly fluent grammatical error correction system? a comprehensive evaluation.

Masahiro Kaneko and Naoaki Okazaki. 2023. Reducing sequence length by predicting edit operations with large language models.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, NAACL '03, pages 48–54. Association for Computational Linguistics. 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Langauge Technology (HLT-NAACL 2003) ; Conference date: 27-05-2003 Through 01-06-2003.

Yichong Leng, Xu Tan, Linchen Zhu, Jin Xu, Renqian Luo, Linquan Liu, Tao Qin, Xiangyang Li, Edward Lin, and Tie-Yan Liu. 2021. Fastcorrect: Fast error correction with edit alignment for automatic speech recognition. *Advances in Neural Information Processing Systems*, 34:21708–21719.

Yaniv Leviathan, Matan Kalman, and Yossi Matias. 2023. Fast inference from transformers via speculative decoding.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. 2024. Gpt-4 technical report.

Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. 2015. Librispeech: An asr corpus based on public domain audio books. In *2015 IEEE International Conference on Acoustics,*

*Speech and Signal Processing (ICASSP)*, pages 5206–5210.

Jie Pu, Thai-Son Nguyen, and Sebastian Stüker. 2023. Multi-stage large language model correction for speech recognition.

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.

Emily Reif, Daphne Ippolito, Ann Yuan, Andy Coenen, Chris Callison-Burch, and Jason Wei. 2022. A recipe for arbitrary text style transfer with large language models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 837–848, Dublin, Ireland. 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.

Yu Zhang, Wei Han, James Qin, Yongqiang Wang, Ankur Bapna, Zhehuai Chen, Nanxin Chen, Bo Li, Vera Axelrod, Gary Wang, Zhong Meng, Ke Hu, Andrew Rosenberg, Rohit Prabhavalkar, Daniel S. Park, Parisa Haghani, Jason Riesa, Ginger Perng, Hagen Soltau, Trevor Strohman, Bhuvana Ramabhadran, Tara Sainath, Pedro Moreno, Chung-Cheng Chiu, Johan Schalkwyk, Françoise Beaufays, and Yonghui Wu. 2023. Google usm: Scaling automatic speech recognition beyond 100 languages.