Efficient On-Device Text Simplification for Firefox with Synthetic Data Fine-Tuning

Pablo Romero[1], Zihao Li[2], Matthew Shardlow[1],
[1]Manchester Metropolitan University, [2]University of Manchester,
Correspondence: pablo2004romero@gmail.com, jeremy.li@manchester.ac.uk, m.shardlow@mmu.com

Abstract

This work presents a system for on-device text simplification that enables users to process sensitive text without relying on cloud-based services. Through the use of quantization techniques and a novel approach to controllable text simplification, we reduce model size by up to 75% with minimal performance degradation. Our models demonstrate efficient state-of-the-art results using a synthetic dataset of 2,909 examples, outperforming prior work trained on 300K examples. This efficiency stems from: (1) a single control token strategy that precisely targets specific reading levels, (2) a multi-level training approach that exposes models to transformations from multiple source complexity levels, and (3) individual models that dedicate full parameter capacity to specific reading level transformations. Our best models achieve up to 82.18 BLEU (at the Advanced level) and 46.12 SARI (at the Elementary level) on standard benchmarks, with performance preserved even after aggressive quantization. This work is implemented as a collaboration with the Mozilla AI team to processes text entirely locally, ensuring sensitive information never leaves the user's device. We have a demonstration video[1] and a web demo available at: https://pablorom2004.github.io/Simplification-Web-Demo/

1 Introduction

Text simplification aims to reduce textual complexity while preserving essential meaning, thereby improving accessibility for a broad range of readers (Alva-Manchego et al., 2021). Today, sequence-to-sequence neural models provide state-of-the-art results, but many existing solutions require server-side processing, raising concerns about data privacy and latency when processing sensitive content like medical information or legal documents.

Our work addresses these issues by introducing an on-device text simplification system with two core innovations. First, we implement a single control token strategy instead of relying on multiple complexity metrics. This approach departs from previous work (Li et al., 2022) that employed multiple tokens to represent various linguistic features.

Second, we create a high-quality synthetic dataset comprising only 2,909 examples generated by carefully prompting large language models. Despite its relatively small size, this dataset outperforms the 300K-example WikiLarge corpus (Zhang and Lapata, 2017) on standard benchmarks, highlighting that data quality can far outweigh quantity for efficient model training.

We deploy our models locally in the browser via *transformers.js*[2], ensuring that all data processing happens directly on the user's device. Experiments on standard benchmarks show that our models rival or outperform previous approaches, and maintain their quality even when quantized for efficient on-device operation.

2 Related Work

Text simplification has progressed from rule-based approaches (Elhadad and Sutaria, 2007; Yatskar et al., 2010; Biran et al., 2011) to data-driven methods leveraging parallel corpora (Surya et al., 2019; Martin et al., 2020; Omelianchuk et al., 2021; Martin et al., 2022). For comprehensive overviews of recent developments, see (North et al., 2025). Pre-trained language models like BART (Lewis et al., 2019) have enabled more fluent and faithful simplifications.

Control tokens for controllable text simplification were first introduced by (Scarton and Specia, 2018), with (Martin et al., 2020) and (Spring et al., 2021) further expanding on controllable simplification. (Li et al., 2022) explored different control

[1]https://youtu.be/TzmaxnARMzg

[2]https://huggingface.co/docs/transformers.js

Proceedings of the Fourth Workshop on Text Simplification, Accessibility and Readability (TSAR 2025), pages 105–115

token configurations and found that tokens representing distinct aspects of complexity (e.g., dependency tree depth, word rank, and length ratio) could effectively guide generation. However, their approach used multiple tokens simultaneously, potentially creating competition for model attention and parameter space. Our work builds on this line of research by using a single, level-based control token.

Synthetic data generation using large language models has emerged as a promising direction for low-resource NLP tasks (Wang et al., 2023). Recent work has demonstrated that high-quality synthetic data can match or exceed naturally collected data for various applications (Yang et al., 2023). Our work emphasizes quality and diversity over quantity, showing that careful prompt engineering can produce highly effective training examples.

On-device NLP has gained traction with growing privacy concerns and the need for offline capability. Mozilla's Firefox Translations project (Mozilla NLP Team, 2023) pioneered browser-based machine translation using ONNX format (Foundation, 2017) (Open Neural Network Exchange, an open standard for machine learning deployment) and WebAssembly. Model compression techniques like distillation and quantization (Jain, 2022) have been essential for deploying models in resource-constrained environments. Our work extends these approaches to text simplification, demonstrating successful deployment with minimal performance loss.

3 Model

3.1 Synthetic Dataset Creation

We created a synthetic dataset of 2,909 examples, each with three levels of simplification: Elementary, Secondary, and Advanced. The dataset generation involved a two-stage approach:

First, we developed a detailed prompt[3] using Claude 3.5 Sonnet (Anthropic, 2024), specifically describing task requirements, formatting, and quality expectations. This meta-prompting approach, asking one language model to create prompts for another, represents a valuable technique for data creation. Our prompt included specific instructions for creating sentences at different reading levels, with clear definitions for each simplification level:

- **ELEMENTARY**: Uses very simple words and straightforward structure. Suitable for 5th-7th grade.
- **SECONDARY**: Simple vocabulary. Suitable for 8th-12th grade.
- **ADVANCED**: Keep as one sentence but slightly simpler than complex.

The resulting prompt was then used with OpenAI's o1 model (OpenAI, 2024) (a large multimodal model with strong reasoning capabilities) to generate 2,284 synthetic examples (an example is the complex sentence and the three levels of simplification). We chose the o1 model due to its large context window and reasoning capabilities, which helped ensure high-quality, diverse examples. The prompt instructed the model to generate content across various topics (news, technology, health, education) that Firefox users might encounter during web browsing. We enhance model training by including multiple examples (e.g., training the Elementary model on Complex → Elementary, Advanced → Elementary, and Secondary → Elementary inputs).

To ground our dataset in established benchmarks, we supplemented the synthetic data with 625 examples derived from the WikiLarge training set, where we extracted complex sentences and used our o1 prompt[4] to generate three simplified versions. This hybrid approach balances novel generation with grounding in established datasets. The dataset can be found here: https://github.com/pabloRom2004/Simple-Synthetic-Dataset

3.2 Control Token vs. Separate Models Approaches

We explored two distinct approaches to modeling reading level control:

3.2.1 Control Token Model

For the Control Token Model approach, we trained a single BART-base model with a prepended control token (e.g., <LEVEL_ELEMENTARY>, <LEVEL_SECONDARY>, or <LEVEL_ADVANCED>) indicating the desired reading level. Unlike previous approaches that used multiple control tokens for different aspects of simplification (Li et al., 2022), our approach uses a single token for the entire simplification level.

[3] https://github.com/pabloRom2004/Simple-Synthetic-Dataset/blob/main/Promt1.txt

[4] https://github.com/pabloRom2004/Simple-Synthetic-Dataset/blob/main/Promt2WikiLarge.txt

This single token strategy simplifies the control mechanism and allows the model to learn a more direct mapping between the token and the desired output style. For example, rather than specifying multiple complexity dimensions such as *dependency tree depth*, *word rank*, *levenshtein distance* and *length ratio*, we simply use <LEVEL_ELEMENTARY> to indicate the overall desired simplification level, in line with the main takeaway from the bitter lesson[5], a principle in AI research suggesting that general methods leveraging computation ultimately outperform human-engineered approaches.

3.2.2 Individual Models

For the Individual Models approach, we trained three separate models, each dedicated to one target reading level. This approach exposes models to transformations from multiple source complexity levels during training. The Elementary model is trained using Complex → Elementary, Secondary → Elementary, and Advanced → Elementary transformation pairs. Similarly, the Secondary model incorporates Complex → Secondary and Advanced → Secondary examples, while the Advanced model focuses only on Complex → Advanced transformation pairs. This multi-level training strategy provides each model with a richer set of transformation patterns, helping it learn more robust simplification strategies by observing how text at different complexity levels can be transformed to its target level.

From a theoretical perspective, individual models allocate the entire parameter space to learning one specific level of transformation, avoiding the parameter competition that might occur in a single model trying to learn multiple transformation levels simultaneously. This parameter efficiency becomes particularly important in smaller models like BART-base, where the capacity to represent multiple complex transformations may be limited.

3.3 Training Configuration

We used a BART-base sequence-to-sequence model (Lewis et al., 2019) as our foundation, with the following hyperparameters: 8 epochs, batch size of 8, learning rate of 1e-4 with Adam optimizer, weight decay of 0.01, and 10% warmup steps. The dataset was split using a 99/0.8/0.2 ratio for train/test/validation. This high train percentage was chosen because we evaluated on the standard ASSET Test Set as our benchmark rather than using our own test set, making it possible to allocate more data for training.

Because of the relatively small dataset size, training on an NVIDIA 4080 Super took only approximately 30 minutes, highlighting the computational efficiency of our approach. This rapid training time stands in stark contrast to models trained on large datasets like WikiLarge, which can take many hours to train. The efficiency gain comes from both the smaller dataset size and the clear learning signal provided by our high-quality synthetic examples.

3.4 Deployment via *transformers.js*

We deploy our models on-device using HuggingFace's *transformers.js* framework directly adhering to the FireFox documentation[6], which allows running models hosted on HuggingFace[7] with the ONNX format directly in web browsers. Our deployment pipeline involves two key steps:

We convert our trained models to ONNX format for compatibility with a script provided by *transformers.js*. This conversion is essential for allowing the models to run in standard web browsers without specialized hardware acceleration. In this same script, quantization is applied to reduce model size and memory footprint. We experimented with both INT8 and 4-bit quantization, which reduced file sizes by up to 75% compared to the full-precision FP-32 model with minimal performance degradation.

The resulting system stores models in the browser's local storage, enabling persistent availability across sessions without repeated downloads. Once downloaded, models can be used completely offline, ensuring privacy and reliability even in disconnected environments.

4 Results

4.1 Evaluation Setting

We evaluate on two widely-used text simplification benchmarks: The **ASSET Test Set** (Alva-Manchego et al., 2020) and **TurkCorpus** (Xu et al., 2016), these two benchmarks feature human made simplifications of 359 English sentences from Wikipedia, focusing on fluency, meaning preservation, and simplicity. As a relatively recent benchmark, it offers multiple reference simplifications

[5] http://www.incompleteideas.net/IncIdeas/BitterLesson.html

[6] https://firefox-source-docs.mozilla.org/toolkit/components/ml/

[7] https://huggingface.co/collections/pabRomero/firefox-simplification-67d70f0d3dcb47939026303f

per sentence, created through crowdsourcing with detailed guidelines.

For evaluation, we employ multiple complementary metrics:

- **BLEU** (Papineni et al., 2002): Measures n-gram overlap with reference texts, capturing fluency and preservation of meaning
- **SARI** (Xu et al., 2016): Focuses on evaluating Add, Delete, and Keep operations compared to references, specifically designed for text simplification evaluation
- **BERTScore** (Zhang et al., 2020): Captures semantic similarity using contextual embeddings, offering a more nuanced measure of meaning preservation
- **LENS** (Maddela et al., 2023) a learnt evaluation metric for simplicity assessment
- **SALSA** (Heineman et al., 2023): An edit-level simplification evaluation metric

4.2 Distillation and Quantization

Table 1 shows that quantized models (INT8 or BNB-4) lose minimal performance compared to FP-32 while reducing size by up to 75%. Notably, the INT8 model (136MB) achieves the highest readability scores (LENS, SALSA) despite its smaller size, while BNB-4 outperforms on semantic preservation metrics (BERT-P/R/F1). These results suggest quantization may act as beneficial regularization for certain simplification aspects, making these compressed models ideal for resource-constrained environments without sacrificing quality. The quantization of the models was performed using the framework provided by transformers.js.

4.3 Model Performance on ASSET Test Set

Table 2 compares our approaches with a baseline BART model (Li et al., 2022) trained on WikiLarge.

Both our control token model and individual models outperform the baseline trained on WikiLarge across all metrics, despite our synthetic dataset containing 100 times fewer examples (300,000 vs. 2,909). This striking result challenges conventional wisdom, suggesting that a small set of high-quality synthetic examples can be more effective than a large corpus of lower-quality or less focused examples. The individually trained models

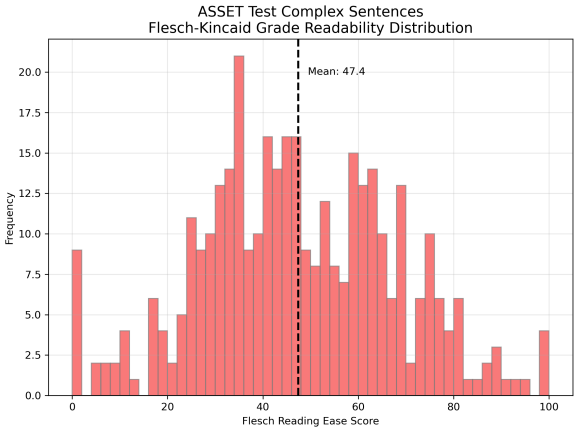

Figure 1: Distribution of Flesch-Kincaid readability scores for original ASSET Test Set sentences. Mean score is around 47.4, reflecting moderate complexity.

outperform the control token model and the baseline, showing the efficacy of specialization into a single, well-defined task for these small language models like BART.

4.4 Readability Analysis

Figure 1 shows the distribution of Flesch-Kincaid readability scores for the original ASSET Test Set sentences. These scores range from 0-100, with higher scores indicating easier readability. Figure 2 shows the distribution of Flesch-Kincaid readability scores of the individual models after simplification.

Figure 3 visualizes how our Individual Models transform text readability according to the Flesch-kincaid score. Each point represents a sentence, with readability score of the original text on the x-axis and the readability of the simplified text on the y-axis. Points above the diagonal line (y=x) indicate simplification; points below it show increased complexity.

This visualization confirms that our three models effectively target distinct reading levels, with clear separation in their simplification behaviours. The Advanced level model stays around the y=x line, which means that the model is generally just re-writing the sentences in a similar level of complexity just with different structure, potentially helping a user understand the sentence once it is re-worded. The Elementary level model shows clear simplification from input to output, nearly all of the examples show clear improvements in their Flesch-Kincaid scores which shows great simplification ability from the model. While we acknowledge that Flesch-Kincaid and other automatic readabil-

Dataset	Model	BLEU	SARI	BERT-P	BERT-R	BERT-F1	LENS	SALSA	Size
ASSET	FP-32	**51.17**	42.07	0.682	0.667	0.657	58.80	68.88	540MB
	INT8	49.96	42.06	0.677	0.668	0.657	**59.19**	**69.86**	136MB
	BNB-4	51.07	**42.12**	**0.688**	**0.674**	**0.665**	58.66	68.86	212MB

Table 1: Performance comparison of BART model variants with different quantization levels on text simplification tasks.

Model	Level	BLEU	SARI	BERT-P	BERT-R	BERT-F1	LENS	SALSA
Baseline	–	51.17	42.07	0.682	0.667	0.657	58.80	68.88
Control	Elem	57.25	43.21	0.728	0.701	0.701	68.97	75.42
Control	Sec	62.98	41.76	0.766	0.750	0.745	61.21	69.83
Control	Adv	52.96	40.32	0.680	0.699	0.678	50.49	65.03
Indiv.	Elem	58.41	**46.12**	0.754	0.747	0.737	**71.39**	**77.54**
Indiv.	Sec	72.81	42.19	0.828	0.828	0.817	62.46	68.91
Indiv.	Adv	**82.18**	35.82	**0.870**	**0.877**	**0.866**	59.42	65.32

Table 2: Model performance on the ASSET Test Set. Individual Models outperform both the baseline and the Control Token model.

ity metrics have known limitations in evaluating text simplification quality (Alva-Manchego et al., 2021), we present these scores as exploratory indicators of relative complexity changes across our models rather than definitive measures of simplification success.

4.5 Example Simplifications

Table 4 in the Appendix presents selected examples from our models, highlighting successes and challenges across different reading levels.

These examples highlight both the strengths of our models (effective simplification at appropriate levels) and areas for improvement (maintaining factual accuracy and avoiding unnecessary transformations).

5 Prototype Implementation

We developed a web-based prototype that demonstrates our text simplification models operating directly in the browser. The implementation uses *transformers.js*, for all processing locally, ensuring privacy by keeping sensitive text on the user's device. The source code is available at `https://github.com/pabloRom2004/Simplification-Web-Demo`.

5.1 Technical Architecture

The prototype follows a fully client-side architecture, operating entirely within the browser without server-side processing. The core components include:

- **Model Management**: Handles downloading, storing, and loading of quantized ONNX models
- **Text Processing**: Implements sentence splitting for input paragraphs, tokenization, and recombination
- **Inference Pipeline**: Configures and executes the simplification models
- **Readability Analysis**: Calculates Flesch-Kincaid scores for original and simplified text
- **Visualization**: Provides interactive display of simplification results with sentence mapping

When a user first visits the application, they select a quantization level (FP32, INT8, or BNB-4) based on their device capabilities and memory constraints. They can then download one or more models at their chosen reading levels. Once downloaded, models persist in the browser's local storage, eliminating the need for re-downloading in future sessions.

For longer texts, we implement a sentence-splitting algorithm that identifies sentence boundaries while accounting for common abbreviations

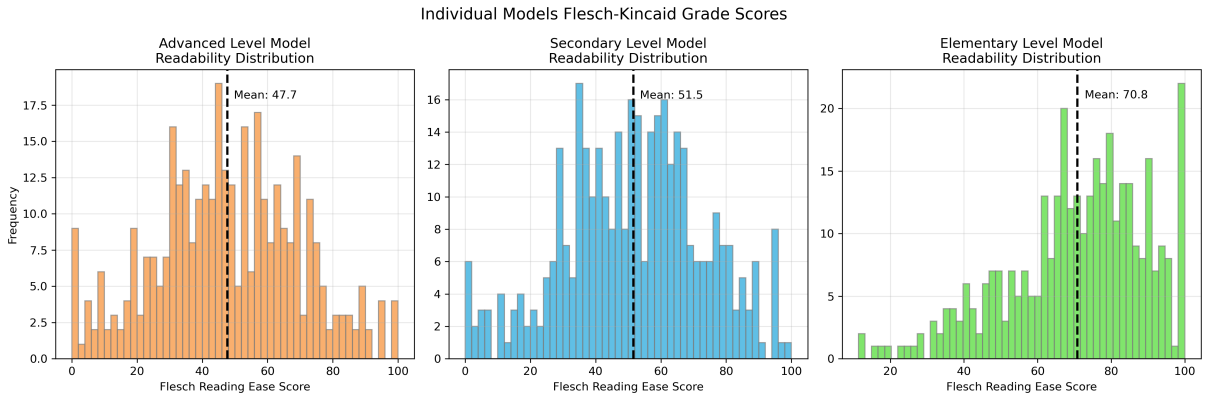

Figure 2: Distribution of Flesch-Kincaid readability scores for individual models on the ASSET Test Set sentences.

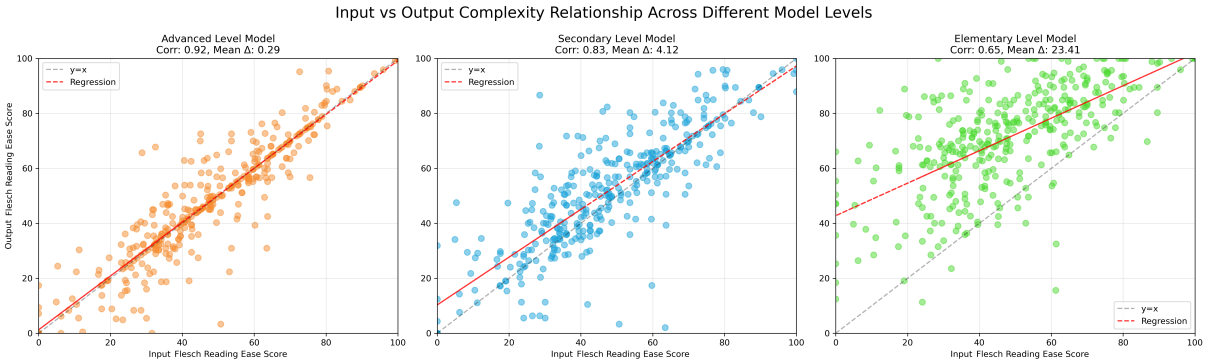

Figure 3: Scatter plot of original (x-axis) vs. simplified (y-axis) readability for the Individual Models. Points above the diagonal line represent simplification, while points below it indicate increased complexity.

and special cases. Each sentence is processed independently before being recombined, allowing efficient handling of paragraphs without exceeding browser memory constraints.

6 Discussion

Our experiments reveal several key insights with broader implications. First, our small but high-quality synthetic dataset (2,909 examples) outperforms the WikiLarge corpus (300K examples), challenging conventional wisdom about data requirements for fine-tuning. This finding suggests that pre-trained models already possess substantial linguistic knowledge and primarily need clear, unambiguous direction rather than extensive examples. Our synthetic data succeeds by precisely pointing the model toward the desired task through concise, well-crafted examples that demonstrate the exact transformation patterns required. The clarity and quality of this directional signal prove significantly more valuable than quantity, indicating that carefully engineered LLM prompts can create highly effective training data for a wide range of downstream NLP tasks.

Second, our single control token strategy demonstrates that simplicity can outperform complexity in control mechanisms. By using a single token that directly indicates the target reading level rather than multiple tokens representing different complexity features, we reduce potential parameter competition and make training more efficient for smaller models.

Third, our finding that individually trained models outperform the control token approach highlights the importance of parameter efficiency in smaller architectures. By dedicating the entire parameter space to learning one specific transformation, these models develop more robust simplification strategies for their target reading levels.

Finally, the impressive stability of performance across quantization levels (with size reductions up to 75%) indicates that many NLP tasks may not require full floating-point precision. The INT8 model's superior performance on readability metrics despite its smaller size suggests that quantization may actually function as beneficial regularization for certain aspects of text simplification.

While effective for sentence-level simplification,

we found extending to longer contexts or domain-specific text challenging for our BART-base models, suggesting larger architectures may be needed for these scenarios.

7 Conclusion

This work presents an on-device text simplification approach using synthetic data and model quantization that processes text locally in Firefox browsers. Our contributions include: (1) demonstrating synthetic LLM-generated data can outperform much larger human-annotated datasets, (2) showing specialized models outperform control token approaches for smaller architectures, and (3) providing a privacy-preserving implementation with state-of-the-art quantization techniques that together enable efficient and private language technologies.

Limitations

Our approach has several limitations. Our reliance on synthetic data, while effective, may miss certain nuances of human-authored simplifications. Our current implementation emphasizes sentence-level simplification rather than document-level coherence, potentially creating local optimizations that do not maintain global coherence in longer texts. We have also not conducted extensive human evaluation, which would be valuable for assessing subjective aspects of simplification quality that automatic metrics may not capture.

Future work will incorporate structured human evaluation with university students to validate our findings beyond automatic metrics, and explore extending the approach to longer contexts and additional languages.

Lay Summary

Reading complex text online can sometimes be too hard to read for some users, especially when encountering technical articles, legal documents, or medical information. This work presents a system that simplifies difficult text directly in your web browser without sending your data to external servers, protecting your privacy.

Our approach makes two key innovations. First, instead of training our models on hundreds of thousands of examples like previous work, we created just 2,909 high-quality examples by carefully prompting advanced AI systems to generate simplified versions of sentences at three reading levels: Elementary (suitable for middle school), Secondary (suitable for high school), and Advanced (slightly simplified but maintaining sophistication). Surprisingly, this small, carefully crafted dataset outperformed much larger datasets, demonstrating that quality matters more than quantity.

Second, we made these models small enough to run in a web browser by compressing them to 25% of their original size while maintaining performance. This means users can simplify sensitive text like medical records or legal documents without that information ever leaving their device.

The system offers three simplification levels, allowing users to choose how much simplification they need. For example, a medical article about "dextromethorphan occurring as a white powder in its pure form" might become "Dextromethorphan is a white powder" at the Elementary level, while maintaining more detail at higher levels.

We built this as a working web demo and collaborated with Mozilla's Firefox team to integrate it into the browser. The models work entirely offline once downloaded, making simplified reading accessible even without an internet connection. This work shows that privacy-preserving, accessible language technology can be both practical and powerful.

References

Fernando Alva-Manchego, Louis Martin, Antoine Bordes, Carolina Scarton, Benoît Sagot, and Lucia Specia. 2020. ASSET: A dataset for tuning and evaluation of sentence simplification models with multiple rewriting transformations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4668–4679, Online. Association for Computational Linguistics.

Fernando Alva-Manchego, Carolina Scarton, and Lucia Specia. 2021. The (un)suitability of automatic evaluation metrics for text simplification. *Computational Linguistics*, 47(4):861–889.

Anthropic. 2024. Introducing claude 3.5 sonnet. Accessed on March 14, 2025.

Or Biran, Samuel Brody, and Noémie Elhadad. 2011. Putting it simply: a context-aware approach to lexical simplification. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 496–501.

Noemie Elhadad and Komal Sutaria. 2007. Mining a lexicon of technical terms and lay equivalents. In *Biological, translational, and clinical language processing*, pages 49–56.

Linux Foundation. 2017. ONNX | onnx.ai. https://onnx.ai/. [Accessed 14-03-2025].

David Heineman, Yao Dou, Mounica Maddela, and Wei Xu. 2023. Dancing between success and failure: Edit-level simplification evaluation using SALSA. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 3466–3495, Singapore. Association for Computational Linguistics.

Shashank Mohan Jain. 2022. Hugging face. In *Introduction to transformers for NLP: With the hugging face library and models to solve problems*, pages 51–67. Springer.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

Zihao Li, Matthew Shardlow, and Saeed Hassan. 2022. An investigation into the effect of control tokens on text simplification. In *Proceedings of the Workshop on Text Simplification, Accessibility, and Readability (TSAR-2022)*, pages 154–165, Abu Dhabi, United Arab Emirates (Virtual). Association for Computational Linguistics.

Mounica Maddela, Yao Dou, David Heineman, and Wei Xu. 2023. LENS: A learnable evaluation metric for text simplification. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 16383–16408, Toronto, Canada. Association for Computational Linguistics.

Louis Martin, Éric de la Clergerie, Benoît Sagot, and Antoine Bordes. 2020. Controllable sentence simplification. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4689–4698, Marseille, France. European Language Resources Association.

Louis Martin, Angela Fan, Éric de la Clergerie, Antoine Bordes, and Benoît Sagot. 2022. MUSS: Multilingual unsupervised sentence simplification by mining paraphrases. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 1651–1664, Marseille, France. European Language Resources Association.

Mozilla NLP Team. 2023. Mozilla translations: Open-source neural translation in the browser. In *Proceedings of Machine Translation Summit*.

Kai North, Tharindu Ranasinghe, Matthew Shardlow, and Marcos Zampieri. 2025. Deep learning approaches to lexical simplification: A survey. *Journal of Intelligent Information Systems*, 63:111–134.

Kostiantyn Omelianchuk, Vipul Raheja, and Oleksandr Skurzhanskyi. 2021. Text Simplification by Tagging. In *Proceedings of the 16th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 11–25, Online. Association for Computational Linguistics.

OpenAI. 2024. Learning to reason with LLMs. https://openai.com/index/learning-to-reason-with-llms/. [Accessed 14-03-2025].

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, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Carolina Scarton and Lucia Specia. 2018. Learning simplifications for specific target audiences. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 712–718, Melbourne, Australia. Association for Computational Linguistics.

Nicolas Spring, Annette Rios, and Sarah Ebling. 2021. Exploring German multi-level text simplification. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2021)*, pages 1339–1349, Held Online. INCOMA Ltd.

Sai Surya, Abhijit Mishra, Anirban Laha, Parag Jain, and Karthik Sankaranarayanan. 2019. Unsupervised neural text simplification. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2058–2068, Florence, Italy. Association for Computational Linguistics.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. Self-instruct: Aligning language models with self-generated instructions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*.

Wei Xu, Courtney Napoles, Ellie Pavlick, Quanze Chen, and Chris Callison-Burch. 2016. Optimizing statistical machine translation for text simplification. *Transactions of the Association for Computational Linguistics*, 4:401–415.

Rui Yang, Haotian Lin, Cheng Wang, and Hao Qian. 2023. Gpt4tools: Teaching large language models to use tools via self-instruction. In *Proceedings of the 40th International Conference on Machine Learning*.

Mark Yatskar, Bo Pang, Cristian Danescu-Niculescu-Mizil, and Lillian Lee. 2010. For the sake of simplicity: Unsupervised extraction of lexical simplifications from wikipedia. *arXiv preprint arXiv:1008.1986*.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. *Preprint*, arXiv:1904.09675.

Xingxing Zhang and Mirella Lapata. 2017. Sentence simplification with deep reinforcement learning. *arXiv preprint arXiv:1703.10931*.

A Additional Results and Examples

Model	Level	BLEU	SARI	BERT-P	BERT-R	BERT-F1	LENS	SALSA
Baseline	–	46.94	36.87	0.678	0.613	0.636	58.65	68.88
Control	Elem	51.25	37.63	0.720	0.652	0.679	68.67	75.42
Control	Sec	59.87	38.28	0.778	0.726	0.745	61.62	69.83
Control	Adv	52.75	36.76	0.693	0.682	0.681	51.22	65.03
Indiv.	Elem	50.60	**39.48**	0.733	0.693	0.704	**70.89**	**77.54**
Indiv.	Sec	71.45	40.37	0.846	0.814	0.825	63.00	68.91
Indiv.	Adv	**84.92**	37.63	**0.912**	**0.894**	**0.900**	61.10	65.32

Table 3: Model performance on TurkCorpus.

B Dataset Creation Prompts

The prompts used to generate our synthetic dataset (including the WikiLarge-based simplifications) are available at: https://github.com/pabloRom2004/Simple-Synthetic-Dataset

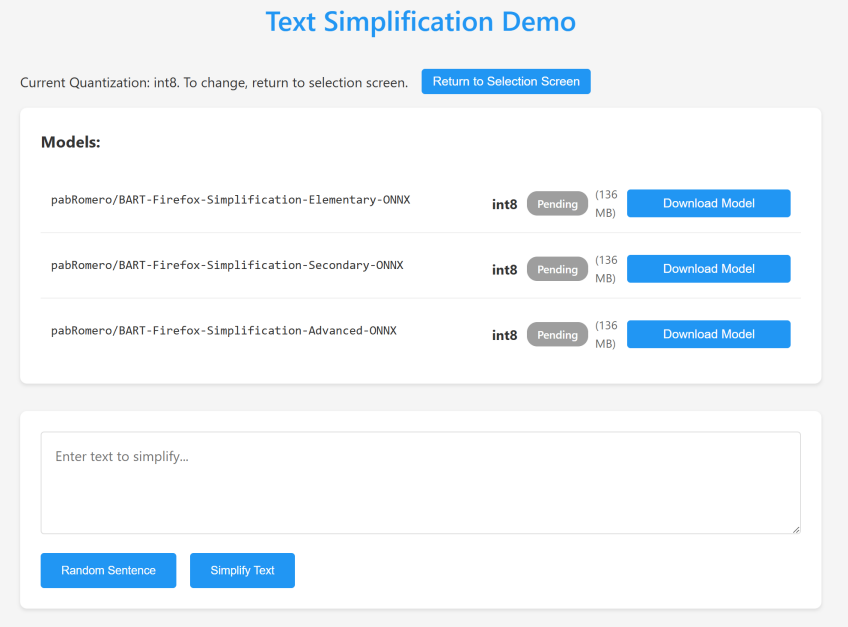

Figure 4: Screenshot of the web-based prototype interface showing model selection, text input, and results display with color-coded sentence mapping.

Type	Example
Elementary (Success)	
Original	The SAT Reasoning Test (formerly Scholastic Aptitude Test and Scholastic Assessment Test) is a standardized test for college admissions in the United States.
Simplified	The SAT is a test that students can take to get a good grade. It's used to help them find colleges.
Elementary (Failure)	
Original	In its pure form, dextromethorphan occurs as a white powder.
Simplified	Dextromethorphan is a white powder.
Secondary (Success)	
Original	Below are some useful links to facilitate your involvement.
Simplified	Here are some helpful resources to help you get involved.
Secondary (Failure)	
Original	Protests across the nation were suppressed.
Simplified	Demonstrations across the nation were peaceful.
Advanced (Success)	
Original	Minor planet designations are number-name combinations overseen by the Minor Planet Center, a branch of the IAU.
Simplified	Minor planets are classified by the Minor Planet Center, a branch of the IAU.
Advanced (Failure)	
Original	About 95 species are currently accepted.
Simplified	Approximately 95 species are currently accepted.

Table 4: Example simplifications from different models and reading levels. "Failures" are cases where the model either fails to maintain factual accuracy (e.g., changing "suppressed" to "peaceful"), performs minimal transformation when simplification is needed, or unnecessarily complicates simple text (e.g., "about" to "approximately").

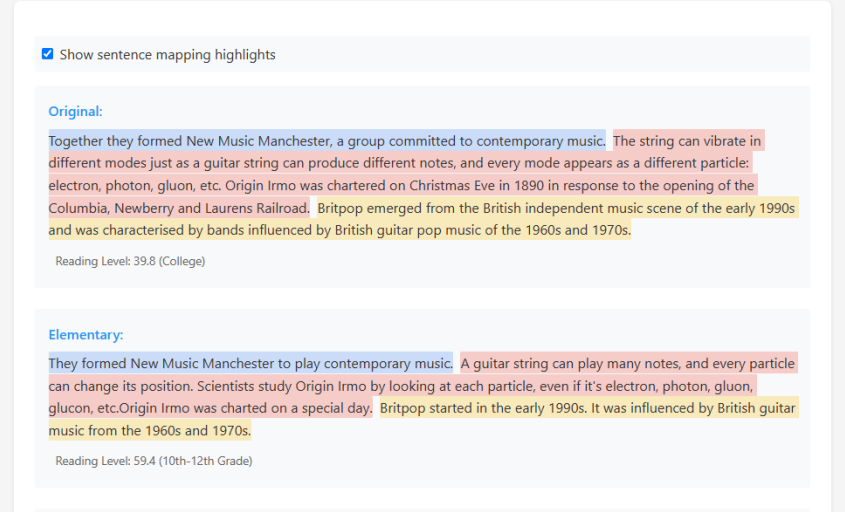

Figure 5: Screenshot of the web-based prototype interface showing sentence splitting from a paragraph, each sentence is individually processed by the model, then re-constructed.