ABSTRACT

Title of dissertation: COMPLEXITY CONTROLLED NATURAL LANGUAGE GENERATION

Sweta Agrawal, Doctor of Philosophy, 2023

Dissertation directed by: Professor Marine Carpuat
Department of Computer Science

Generating text at the right level of complexity for its target audience so it can be easily understood by its target audience has the potential to make information more accessible to a wider range of people, including non-native speakers, language learners, and people who suffer from language or cognitive impairments. For example, a native Hindi speaker learning English might prefer reading a U.S. news article in English or Hindi tailored to their vocabulary and language proficiency level. Natural Language Generation (NLG), the use of computational models to generate human-like text, has been used to empower countless applications – from automatically summarizing financial and weather reports to enabling communication between multilingual communities through automatic translation. Although NLG has met some level of success, current models ignore that there are many valid ways of conveying the same information in a text and that selecting the appropriate variation requires knowing who the text is written for and its intended purpose.

To address this, in this thesis, we present tasks, datasets, models, and algorithms that are designed to let users specify how simple or complex the generated text should be in a given language. We introduce the Complexity Controlled Machine Translation task, where the goal is to translate text from one language to another at a specific complexity level defined

by the U.S. reading grade level. While standard machine translation (MT) tools generate a single output for each input, the models we design for this task produce translation at various complexity levels to suit the needs of different users. In order to build such models, we ideally require rich annotation and resources for supervised training, i.e., examples of the same input text paired with several translations in the output language, which is not available in most datasets used in MT. Hence, we have also contributed datasets that can enable the generation and evaluation of complexity-controlled translations. Furthermore, recognizing that when humans simplify a complex text in a given language, they often revise parts of the complex text according to the intended audience, we present strategies to adopt general-purpose Edit-based Non-Autoregressive models for controllable text simplification (TS). In this framing, the simplified output for a desired grade level is generated through a sequence of edit operations like deletions and insertions applied to the complex input sequence. As the model needs to learn to perform a wide range of edit operations for different target grade levels, we introduce algorithms to inject additional guidance during training and inference, which results in improved output quality while also providing users with the specific changes made to the input text. Finally, we present approaches to adapt general-purpose controllable TS models that leverage unsupervised pre-training and low-level control tokens describing the nature of TS edit operations as side constraints for grade-specific TS.

Having developed models that can enable complexity-controlled text generation, in the final part of the thesis, we introduce a reading comprehension-based human evaluation framework that is designed to assess the correctness of texts generated by these systems using multiple-choice question-answering. Furthermore, we evaluate whether the measure of correctness (via the ability of native speakers to answer the questions correctly using the simplified texts) is captured by existing automatic metrics that measure text complexity or meaning preservation.

COMPLEXITY CONTROLLED NATURAL LANGUAGE GENERATION

by

Sweta Agrawal

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2023

Advisory Committee:
Professor Marine Carpuat, Chair
Professor Philip Resnik, Dean's Representative
Professor Abhinav Shrivastava, Member
Professor Jordan Boyd-Graber, Member
Dr. Ani Nenkova, Member

Dedicated to my loving grandmother, Late Mrs. Rukimini Devi.

Acknowledgments

I am grateful for all the support I have received during my Ph.D. program from many people and I would like to express my gratitude to all of them who contributed to this journey.

First, I would like to sincerely thank my advisor, Prof. Marine Carpuat, whose feedback and support have been the heart of everything I have contributed to research over the past five years. She has helped me grow as a researcher and as a person. I am grateful for the countless hours she has dedicated to reviewing my work, offering valuable insights, and helping me refine my ideas. I would also like to extend my sincere gratitude and appreciation to my thesis committee, Prof. Philip Resnik, Prof. Jordan Boyd-Graber, Prof. Abhinav Shrivastava, and Dr. Ani Nenkova for their invaluable guidance, expertise, and support throughout the completion of my research work. Their feedback, thought-provoking questions, and suggestions have played a pivotal role in improving the quality and depth of my work.

I have been blessed to work with many researchers throughout my Ph.D. program. Many thanks to Julia Kreutzer, George Foster, Colin Cherry, Markus Freitag, Joel Tetreault, Marjan Ghazvinized, Luke Zettlemoyer, Mike Lewis, and Chunting Zhou for the internship and collaboration opportunities. I would also like to thank Dr. Ge Gao and Dr. Niloufar Salehi for introducing me to the world of Human-Computer Interaction.

I am forever indebted to Eleftheria Briakou and Pranav Goel. From countless hours of research discussions to providing emotional support during crisis situations to sharing every single triumph and setback, no words of gratitude are going to do justice to what it meant to have their support. From the many many things I have learned from them, they have also taught me what it means to do research and to care about people. Their presence in my life has been a constant reminder that I am not alone in my journey. Their impact on my life is immeasurable, and I will forever be grateful for their presence by my side.

I would also like to express my deepest gratitude to Yoo Yeon Sung, Yogarshi Vyas, Alexander Hoyle, Weijia Xu, Xing Niu, Maharshi Gor, Rupak Sarkar, Navita Goyal, Neha Srikanth, Chenglei Si, Conner Balmer, Sathvik Nair, Elijah Rippeth, Michelle Yuan, Kiante Brantley, Shi Feng, Khanh Nguyen, Aquia Richburg, Trista Cao, Abhilasha Sancheti, Pedro Rodriguez, Sudha Rao, Joe Barrow, Shohini Bhattasali, and all the members of the CLIP lab. Every one of them has played an integral role in creating a vibrant and collaborative environment. The synergy of discussions, shared ideas, and collective efforts have not only enhanced the quality of my research but also made the lab a place of inspiration and support. I am grateful for the friendship, encouragement, and countless scientific and personal insights that they have generously shared. I would also like to thank Brian Thompson and Huda Khayrallah for guiding me through tough career decisions and it has been an absolute joy working on research projects with Nikita Mehandru and Millicent Li.

A shoutout to my dearest friend, Shlok Mishra, who has always managed to make me laugh even in the toughest situations and ensured that I am keeping good care of my mental and physical well-being. Sharing conference and internship experiences with Aditya Gupta would enrich it incredibly. Thank you for amusing me throughout this delightful decade. And to all my friends who have always helped me see the silver lining through the toughest times: Susmija Jabireddy, Anshul Shah, Kamal Gupta, Ketul Shah, Aman Gupta, Ameya Patil, Samyadeep Basu, Sneha Gathani, Belen Saldias Fuentes, Avira Rajesh, Bhanu Motluru, Saikanth Dacha, Yogesh Balaji, Jitendra Choudhary, Girnar Goyal, Rishabh Jain, and Vatsal Sanjay. Your love, understanding, and belief in my abilities have been my constant source of strength and I am deeply grateful for the same.

Last, but not the least, to my parents, thank you for the unwavering love you have showered upon me since the day I was born, and to my siblings (Annapurna Agrawal, Anjali Agrawal, and Juhi Jain) who have always stood together with me, supporting, motivating, and cheering.

Table of Contents

Chapter 1: Introduction

Natural Language Generation (NLG) is a sub-field in Natural Language Processing that aims at automatically producing natural language texts to facilitate communication and understanding amongst humans or between humans and computers. NLG technologies have been used to empower countless applications ranging from converting and summarizing financial or weather reports into digestible textual content to enabling communication between multilingual communities by automatically translating a text from one language to another. One of the core applications of NLG technology is to present information to people in a manner they find easy to *comprehend*. In text generation, for example, this can be realized by generating the output in a language that is known to the user in simple terms and in a user-friendly way. Generating text at the right level of *complexity* has the potential to make machine-generated outputs more useful and accessible to many users including non-native speakers, language learners (Petersen and Ostendorf, 2007; Allen, 2009), people who suffer from language impairments (Carroll et al., 1999; Canning et al., 2000; Inui et al., 2003) and for readers lacking expert knowledge of the topic discussed (Elhadad and Sutaria, 2007; Siddharthan and Katsos, 2010). For example, a five-year-old child might have difficulty understanding "Coronavirus disease (COVID-19) is an infectious disease caused by the SARS-CoV-2 virus.", but they can potentially understand the simpler rewrite of the same sentence, "Coronavirus is a type of disease and it can make you very sick.". Readers can also benefit from reading text that is better targeted to their needs by being easier to read than the original (Crossley et al., 2014; Yang et al., 2021). A native Hindi speaker learning

English might prefer reading a news article in English or Hindi tailored to their vocabulary and literacy level. Generating text appropriate in their pragmatic context and the correct language is fundamental in conveying the meaning of the original information (Hovy, 1987).

1.1 Thesis Statement

In this thesis, we **argue that natural language generation of text should be tailored toward specific audiences or user groups in a manner that facilitates its understanding** and **design models that can enable audience-specific text generation.** Using machine translation as a test case, we first contribute evidence that the standard sequence-to-sequence models used for many text generation tasks are only able to generate a small fraction of the diverse outputs produced by humans. Next, we introduce a series of tasks, models, and training regimes that help users specify how simple or complex the generated text should be. Finally, we present new protocols to evaluate the correctness of the outputs generated by controllable text generation models.

1.2 Roadmap

We start by surveying the body of literature relevant to the thesis in Chapter 2 and highlight gaps in the existing work to establish and motivate controllable text generation.

In Chapter 3, we first empirically explore whether existing MT systems are capable of capturing the diversity of translations produced by humans. We present two strategies to obtain multiple outputs from a standard neural MT model: we expose the MT model to the different valid variations of the translations that exist for a source text, weighted by their likelihood of being generated by the users. We further filter the candidate translations generated by the MT model using a classifier trained to distinguish valid and incorrect translations. Our evaluation suggests that while standard MT systems can capture users'

preferences and generate multiple translation variations to some extent, they fail to capture the full spectrum of diverse translations generated by humans.

To explore the space of the possible diverse translations more systematically, we introduce tasks, datasets, and models that allow us to capture the intended target audience for a text more directly. The variations in human-generated translations reflect their language and reading proficiency via their choice of vocabulary, sentence structure, and the style of the outputs. **In Chapter 4**, we introduce the task of **Complexity Controlled Machine Translation** (CCMT), where the goal is to *generate a machine-translated output of a source text in a given target language and at an appropriate complexity level* (determined by the reading grade level). While prior work on text complexity has mostly focused on simplifying input in one language, primarily English (Chandrasekar et al., 1996a; Coster and Kauchak, 2011a; Siddharthan, 2014; Xu et al., 2015; Scarton and Specia, 2018a; Kriz et al., 2019; Nishihara et al., 2019), we frame this task in the context of machine translation. We ground the definition of complexity in professionally edited text simplification corpora and create a high-quality Spanish-English (Es-En) dataset that includes examples of Spanish segments paired with several English translations that span a range of reading levels. This dataset is drawn from Newsela, an instructional content platform designed to make education more accessible to students. Unlike regularly occurring bilingual parallel corpora—pairs of sentences in two languages that can be considered translations of each other—our dataset includes multiple translations of the same source text that differ in sentence length and structure, reflecting complex syntactic and lexical simplification operations. We introduce multitask (simplify-and-translate) and pipeline-based (e.g., simplify-then-translate) approaches to address this challenging task. We empirically show that multitask models produce better and simpler translations than pipelines of independent translation and simplification models.

Original: The Mauritshuis museum is **staging an exhibition focusing** on the 17th century self-portraits**, highlighting** the similarities and the differences between **modern-day snapshots and historic works of art.**
Simplified for Grade 7: The Mauritshuis museum is **now set to open an exhibit** on the 17th century self-portraits**. It shows** the similarities and differences between **modern photos and artworks.**
Simplified for Grade 3: The Mauritshuis museum is **going to show self-portraits**.

Table 1.1: Simplified text changes depending on the reading grade level of the target audience. The bold font highlights changes compared to the original version.

Simplifying a text in a given language requires rewriting it so that it is easier to read by the intended audience. In the example 1.1: an originally complex source sentence is split and partially rephrased with simpler language to match the complexity level of a seven-grade student. When simplifying for grade 3, phrases are further simplified, and part of the text is entirely deleted. Ideally, we want to design models for this task with a strong inductive bias toward text editing rather than text generation, as there is a significant overlap between the original and the simplified text. Motivated by the same, Chapter 5 presents the framing of **Controllable Text Simplification (TS) as a text editing task** where the simplified output is generated by iteratively refining a source text through a sequence of edit operations like deletions or insertions. We tailor general-purpose *Edit-based Non-Autoregressive (*NAR*) Sequence-to-Sequence models*, designed for MT to be used for this task. We present inference and training strategies to adapt edit-based models for Controllable TS. Our inference strategy utilizes lexical complexity information derived from data statistics to identify difficult words or phrases in the source text that need to be substituted. And, to enable better learning of the complex structural and lexical transformations by the model, we introduce a new training framework, EDITING CURRICULUM. Our framework exploits the full spectrum of available training samples more effectively by introducing examples of intermediate complexity levels and exposing the model to easy-to-learn edit

operations first, gradually increasing the difficulty of training examples as the model becomes competent. Our strategies improve the quality of the outputs and better match the desired target complexity levels over strong baselines while avoiding rewriting the simplified output from scratch.

An alternative approach prominent in recent work controls the simplicity of generated text by using large-scale pre-trained generative language models and low-level control token values describing the nature of simplification operations to be performed (Martin et al., 2020a). While specifying a desired reading grade level might be more intuitive for lay users, it only provides a weak control over the nature of simplification. On the other hand, controlling the outputs' simplicity by setting several low-level properties (ex: number of words, dependency tree depth) provides a much-finer-grained control but can be cumbersome to set by readers, teachers, or other users. As a result, it remains unclear how to operationalize the control of text simplification in practice. Hence, in Chapter 6, we present simple approaches to automatically predict what low-level control tokens are needed for a given input text and a desired complexity level, based on surface-form features extracted from the source text and the desired complexity level. Our results show that this approach is effective at improving the quality and controlling the degree of simplification in generated outputs based on the automatic evaluation and yields more diverse edit operations than alternative ways of setting control for grade-specific TS.

Prior work, including our own, heavily relies on automatic metrics to evaluate the quality of TS outputs generated by the different models due to a lack of systematic human evaluation methods for assessing controllable TS systems. In Chapter 7, we present a reading-comprehension-based human evaluation framework that is designed to test the correctness of generated text via controllable TS systems. We conduct a thorough evaluation of simplified texts generated by nine automatic controllable text generation models. Our human evaluation show that supervised systems that leverage pre-training knowledge achieves the

highest accuracy on the RC tasks amongst the automatic controllable TS systems. We also show that edit-based models can generate equally correct outputs at different complexity levels compared to generative supervised autoregressive models, further corroborating the efficacy of the presented edit-based models. However, a notable drawback is their inability to preserve the meaning of the original text. Consequently, even the best-performing supervised system struggles with at least 14% of the questions, marking them as "unanswerable" based on the modified content.

In Chapter 8, we conclude by summarizing the contributions and findings of this thesis and present directions for future work.

1.3 Contributions

This dissertation makes the following contributions:

Task Framing Recognizing that there are many valid ways to express the same information in text and that the appropriate variation is dependent upon the target audience, we introduce the Complexity Controlled Machine Translation task that lets the user specify how simple or complex a text should be in a target language. While controlled TS has been studied in a monolingual setting to some extent, its application to translation has not been explored. Furthermore, we present a framing of controllable TS that uses text editing to generate simplified output for a desired complexity level.

Datasets To enable the generation and evaluation of audience-specific text generation, we curate a high-quality Spanish-English (Es-En) dataset that includes examples of Spanish segments paired with several English translations that span a range of reading levels. We additionally collect and release human judgments of the correctness of outputs generated

from a wide variety of controllable text simplification models to facilitate the design and evaluation of automatic metrics for text simplification.

Models We present methods to control the complexity of machine-generated text in both monolingual and cross-lingual settings. Our algorithms are able to leverage the existing training datasets more effectively by either learning to utilize diverse training signals in a multi-task learning paradigm or by using strong inductive bias using task-specific model architecture design.

Evaluation We design and present a new human evaluation framework to assess the correctness of model-generated texts and conduct a thorough human evaluation of both several controllable TS systems. Furthermore, we conduct a meta-evaluation of exiting automatic TS metrics in their ability to capture the correctness of generated texts at the paragraph level.

Chapter 2: Background

The notion of controlling attributes of a text was first discussed in Hovy (1990) with the goal of generating stylistically appropriate variations of the text using a single representation. As when humans generate text, they tailor it to their audience and context and any computer-based text-generating software should be able to generate similar variations of the text using some means of representing the context and the interpersonal goals. While there could be many different and diverse restatements or paraphrases of the original source text, our work focuses on generating the variation of the *source text* in a *target language* that is *tailored toward a user group or a target audience* with the goal of facilitating its understanding and comprehension.

Hence, in this thesis, we focus on two main text generation tasks that take text as input and rewrite it in some new form as output, machine translation (MT) and text simplification (TS). We provide the definition for these tasks, existing datasets, and their evaluation in Section 2.1. We then review approaches and architectures used to model these tasks in Section 2.2. Finally, we discuss the notion of control associated with these natural language generation tasks and approaches proposed in the literature to induce such control in Section 2.3.

2.1 Task Definitions & Evaluation

2.1.1 Machine Translation (MT)

Definition The goal of MT is to generate a translation of a text from one language to another. More formally, an MT model optimizes for generating the most likely translation, $\hat{Y}$, given a source sequence, X, in a target language, L. This requires encoding the meaning of the source text, X, in a representation that conveys the original information and then decoding the same information in the target language as a sequence of tokens, $\hat{Y}$ (Weaver, 1952).

In Translation Studies, the objective of translation is to generate one possible literal rendition of the source text in the target language. However, there is hardly ever only one correct translation, and the choice of the translation is often dependent on the context and the audience (Gengshen, 2003; Milton, 2009). It is, therefore, beneficial that MT systems also generate outputs that better capture the diversity of the set of plausible and semantically or meaning-equivalent translations and are adaptable to the target audience. Yet, research in MT has mostly focused on generating and evaluating a single translation of a source text. We review efforts in the literature to generate diverse MT translations with/without specific properties in Section 2.3.

Datasets Most existing datasets typically include a pair of input-output examples that are translations of each other, i.e., parallel text, and are curated by crawling the web (Resnik, 1998) or aligning texts available in multiple languages (Gale et al., 1994). The amount of resource available for each language vary, and as the web is English-centric, the majority of the available parallel text is too. While this has served as a very important resource for developing MT systems, the outputs generated by these MT systems often lack diversity and are not tailored to user needs. We summarize some of the existing multi-reference

Dataset	Property	Languages	Description
Attribute-labeled			
SATED (Michel and Neubig, 2018)	Author	en-de/fr/es	speaker annotated TED talks roughly containing 271K sentences for each language from 2324 talks.
MuST-SHE (Bentivogli et al., 2020)	Gender	en-es/fr/it	2,136 (audio, transcript, translation) triplets from 273 speakers annotated with gender-related phenomena.
Multi-Reference			
MTC (LDC2002T01)	Paraphrases	en-zh	4-11 sets of human translations for a single set of 100 Mandarin Chinese news articles.
NIST 2002-2008	Paraphrases	zh/ar-en	10,000 Chinese sentences and 10,000 Arabic sentences each with 4 different English translations.
TaPaCo (Scherrer, 2020)	Paraphrases	73 language-pairs	100 paraphrase sets with 200- 250K sentences per language extracted from Tatoeba.
STAPLE (Mayhew et al., 2020)	Paraphrases	en-pt/hu/ja/ko/vi	3k-5k prompts with varying sets of possible translations, weighted and ranked according to actual learner response frequency.
CoCoA-MT (Nădejde et al., 2022)	Formality	en-de/es/hi/ja/it/ru	training and test sets (600) comprising of source segments paired with two contrastive reference translations, one for each formality level.

Table 2.1: MT attribute-labeled and multi-reference datasets.

and attribute-labeled MT datasets and their properties in Table 2.1. None of the existing datasets thus far target readability and comprehension, and to close this gap, we create a new Spanish-English dataset that includes resources for both training and evaluation for this task in Chapter 4.

Evaluation The existence of many equally plausible hypotheses makes both automatic and human evaluation of MT systems very challenging. Human evaluation of machine-translated outputs has been paramount to developing effective MT systems and designing automatic metrics that can provide a close proxy to human judgments. Typically, human annotators are presented with the machine-generated translation with the original source text or a reference translation and asked to rate the generated translation on a scale of one

to five for adequacy (does the translation conveys the correct meaning?) and fluency (is the output good English?) (Lin and Och, 2004) or directly assess the overall quality of the output on a scale of 1-100 (Graham et al., 2016). In order to account for subjectivity in human assessments, typically, multiple annotators are asked to assess the same translation, and the scores are normalized within and across different human annotators. However, different humans might be tolerant of different error types. To address this, an alternative framework for assessing translation quality asks human annotators to label all or most of the error types from a pre-defined set of error categories, which is the basis for the MQM framework (`www.qt21.eu`). While these approaches give a context-independent quality estimate of machine-generated translations, task-based or application-dependent evaluation of MT systems might give a better idea about the usefulness of certain MT systems (Church and Hovy, 2004; Doyon et al., 1999). For example, if the goal of the generated translation is to assist human translators, measuring post-editing effort directly would be useful for human translators to understand whether the MT system is usable (Sanchez-Torron and Koehn, 2016; Zouhar et al., 2021). Likewise, reading comprehension-based human assessment has been used in prior work to test the readability and the adequacy of machine-generated translations (Jones et al., 2005a,b; Berka et al., 2011; Klerke et al., 2015; Scarton and Specia, 2016; Forcada et al., 2018).

The development of automatic metrics to gauge the quality of MT systems has been an active area of research. Due to the high annotation cost of acquiring multi-reference parallel text, MT systems are typically evaluated by comparing the generated hypothesis with a single reference translation using ngram-overlap metrics (e.g., BLEU (Papineni et al., 2002a), METEOR (Banerjee and Lavie, 2005), NIST (Federico and Cettolo, 2007), CHRF (Popović, 2015)) or contextual similarity metrics (e.g., BERTSCORE (Zhang et al., 2020), BLEURT (Sellam et al., 2020), COMET (Rei et al., 2020)). Note that most of these metrics, including BLEU, were designed to be evaluated against a set of references as

different references provide valid variations in style, word choice, and word order (Qin and Specia, 2015), however, this is hardly realized in practice due to the lack of high-quality multi-reference annotated datasets. Freitag et al. (2020) show that different sets of human references can lead to very different system rankings of quality, causing *reference-bias* in MT evaluation. Hence, there have been efforts to design reference-free evaluation metrics (Yankovskaya et al., 2019; Lo, 2019; Zhao et al., 2020; Ma et al., 2019) or to estimate the quality of the generated output by directly comparing the MT generated translation with the original source text (Specia and Shah, 2018; Fomicheva et al., 2020). Automatic task-based evaluation of MT systems uses question answering to test the adequacy of the machine-generated output (Krubiński et al., 2021).

2.1.2 Text Simplification (TS)

Definition Automatic Text Simplification is a task in NLP where the goal is to rewrite, restructure or modify an original text such that it improves the readability of the text while preserving its meaning. Formally, given a source text, X, the goal is to generate an output text, $\hat{Y}$, that is simple and more readable compared to the original source. Due to the ability and potential of TS systems to make texts more accessible to people with various reading or cognitive disabilities, automatically generating a simplified text has been an active area of research in NLP (Stajner, 2021).

However, what makes text simple depends on its target audience (Xu et al., 2015): replacing complex or specialized terms with simpler synonyms might be helpful for non-native speakers (Petersen and Ostendorf, 2007; Allen, 2009) whereas restructuring text into short sentences with simple words might better match the literacy skills of children (Watanabe et al., 2009). Studies of simplification tools for deaf or hard-of-hearing users also show that they prefer lexical simplification to be applied on-demand (Alonzo et al., 2020).

Most prior work in TS focuses on developing models that generate a generic simplified output for a given source text (Xu et al., 2015; Zhang and Lapata, 2017; Alva-Manchego et al., 2020). We contrast this Generic Text Simplification (TS) with Controllable TS which specifies desired output properties either at a high-level such as the target reading grade level for the entire text (Scarton and Specia, 2018a; Nishihara et al., 2019), or low-level, such as the compression ratio or the nature of the simplification operation to use (Mallinson and Lapata, 2019; Martin et al., 2020a; Maddela et al., 2020). Specifying the desired reading grade level might be more intuitive for lay users. However, it provides only weak control over the nature of simplification. In our work, we instead adopt the intuitive framing for Controllable TS where the desired reading grade level is given as input while providing fine-grained control on the generated simplified output by incorporating lexical complexity signals into our model or optimizing the edit actions required (e.g. delete, keep, reorder) to transform a sequence (Chapter 5) or via mapping the grade information to low-level control values using a predictor model (Chapter 6).

Application of TS in NLP Tasks Beyond the application of TS systems in improving text accessibility and readability for humans, they have also been used to improve the performance of downstream NLP tasks such as information extraction (Miwa et al., 2010; Schmidek and Barbosa, 2014), parsing (Chandrasekar et al., 1996b), semantic role labeling (Vickrey and Koller, 2008), machine translation (Gerber and Hovy, 1998; Štajner and Popovic, 2016; Hasler et al., 2017; Štajner and Popović, 2018; Miyata and Tatsumi, 2019; Mehta et al., 2020), among others (Van et al., 2021). As less is understood about the nature and the type of simplification required for a text to be helpful to a machine and our goal is to understand the usability of automatically adapted text simplifications to human audiences or user-groups, we do not explore this research direction in our work.

Dataset	Language	Description
Newsela[2]	EN,ES	News articles in English or Spanish simplified for different reading grade levels (Grade 2-K12) by human experts
Alector (Gala et al., 2020)	FR	79 literary (tales, stories) and scientific (documentary) texts simplified for poor-reading and dyslexic children at morpho-syntactic, lexical, and discourse levels
OneStopEnglish (Vajjala and Lučić, 2018)	EN	189 texts with 3 versions of simplifications targeting L2 learners across three reading levels (beginner, intermediate, advanced)

Table 2.2: Audience-specific TS datasets.

Datasets Research in TS has mainly focused on generating ***a*** simplified output from a complex source text. Many exiting datasets include paired examples of complex-simple sentences or segments like WikiLarge (Zhang and Lapata, 2017), SIMPITIKI (Tonelli et al., 2016), among others.[1] These datasets are generated by automatically aligning sentences from an original source document and a simplified version of the same from the web with no specific target audience. We review target audience-specific TS datasets and their properties in Table 2.2. While Newsela includes high-quality graded text articles, its usage in TS research has been limited to generating *a* simplified output ignoring the grade information with a few exceptions (Scarton and Specia, 2018a; Nishihara et al., 2019). The corpus consists of English articles in their original form, 4 or 5 different versions rewritten by professionals to suit different grade levels as well as optional translations of original and/or simplified English articles into Spanish. The availability of graded articles in both Spanish and English is an unexplored dimension in prior work which we utilize to curate a high-quality complexity-controlled machine translation dataset (Section 4.1).

Evaluation Prior work on human evaluation of automatically generated simplifications rates simplicity, adequacy, and fluency of the outputs at the sentence level on a Likert scale

[1] A thorough review of datsets used for TS can be found in Al-Thanyyan and Azmi (2021a).

of 1-5 (Schwarzer and Kauchak, 2018). Angrosh et al. (2014); Laban et al. (2021) use question answering to test the comprehension of the simplified article or paragraph. Devaraj et al. (2022) show that factual errors often appear in both human and automatically generated simplified texts (at the sentence level) and propose an error taxonomy to account for both the nature and severity of errors. Maddela et al. (2022) introduce RANK & RATE, a human evaluation framework that rates simplifications from several models by leveraging automatically annotated edit operations that are first verified by the annotators and then used to rate the output text on a scale of 0-100 jointly for meaning preservation, simplicity, and fluency. While both approaches provide reliable and useful indications of the quality and correctness of the simplified text at the sentence level, they do not account for the context in which the sentences appear and their impact on the understanding of the overall text. To the best of our knowledge, there is no prior work on audience or context-dependent evaluation of TS systems. We introduce a human evaluation of grade-specific simplification in our completed work in Section 5.4 and aim to further close the gap by directly assessing the correctness of controlled simplification via a human evaluation (Chapter 7).

The most widely used automatic metric for TS evaluation is SARI (Xu et al., 2016). SARI measures lexical simplification based on the words that are added, deleted and kept by the systems by comparing system output against references and against the input sentence. It computes the F1 score for the n-grams that are added (**add-F1**). The model's deletion capability is measured by the F1 score for n-grams that are kept (**keep-F1**) and precision for the n-grams that are deleted (**del-P**) [3] Xu et al. (2016) show that BLEU shows a high correlation with human scores for grammaticality and meaning preservation, and SARI shows a high correlation with human scores for simplicity. Another common way to evaluate simplicity is to use readability metrics (e.g., FLESCH-KINCAID grade level or *Automatic Reading Index* (ARI)) to estimate the average reading level of the generated simplified

[3] https://github.com/cocoxu/simplification

outputs. While this score gives an automatic measure of corpus-level simplification, it does not provide any estimate on whether the generated outputs match a desired target grade level or whether the model over- or under-simplified the source text. Heilman et al. (2008), therefore, propose *Adjacency Accuracy* to measure the proportion of predictions that were within one grade level of the human assigned grade for the given text. Taken together, these automatic metrics provide a way to automatically assess the simplicity of the generated outputs via TS systems.

2.2 Sequence-to-Sequence Models

Neural Sequence-to-Sequence models or encoder-decoder models are used to convert one sequence of tokens (X) to another (Y) and are used for modeling text generation or structured prediction tasks (Sutskever et al., 2014; Neubig, 2017). The encoder generates a compressed or a sequence of latent representations of the input tokens, which is then utilized by the decoder to generate the target sequence autoregressively (AR, one token at a time) or non-autoregressively (NAR, all tokens at once).

Architecture The encoder or the decoder modules typically have recurrent neural networks (Medsker and Jain, 2001; Hochreiter and Schmidhuber, 1997) or transformers (Vaswani et al., 2017) or both (Chen et al., 2018) as the core architecture. In order to enable the model to handle long-range dependencies and use the token level information from the source sequence more effectively, the encoder-decoder models are equipped with an attention mechanism (Bahdanau et al., 2015). This attention mechanism allows the decoder to combine the information from the hidden representation of different source tokens instead of using a single compressed latent representation of the source sentence. The decoder uses a weighted representation of the source tokens using an attention vector that emphasizes the

amount of attention the decoder should place on each of the source tokens. The transformer architecture dispenses the recurrence completely by only relying on attention mechanism and has been indisputably the architecture of choice for modeling most of the NLP tasks and has lead to significant improvements and state-of-the-art results for these tasks.

Inference Given the probability distribution $P(Y|X)$ generated by the models, decoding is typically performed using one of the following approaches:

a) Greedy Decoding: One of the simplest ways to convert a distribution over the vocabulary tokens to a single token is to select the word with the highest probability, i.e. $\hat{Y} = argmax\ P(y_i|y_1, y_2...y_{i-1}, X)$.

b) Beam Search: Beam Search Decoding (Graves, 2012) is used for AR generation and is based on the observation that the most likely tokens at each timestep might not be globally the best outcome at the sequence level. Hence, in beam search, a set of k partial hypotheses is maintained and eventually, the best hypothesis with the highest sentence level probability is selected as the target output. Intuitively larger beam sizes should result in improved output quality but practically, k beyond 4 or 5 decreases performance due to a lack of calibration in the model's probability distribution (Koehn and Knowles, 2017).

c) Top-k Sampling: Sampling from the probability distribution, instead of maximizing the probability can increase output diversity and reduce bias introduced by greedy decoding. However, sampling from the entire distribution often results in disfluent and degenerate outputs. In Top-k sampling, the next word is instead sampled from the top k most probable choices (Fan et al., 2018).

d) Nucleus Sampling: Nucleus sampling (Holtzman et al., 2020) uses the intuition that the probability mass at each timestep is often concentrated on a few vocabulary tokens.

So, instead of relying on a fixed number of top-k tokens, the next token is sampled from a top-p portion of the probability mass.

Training Sequence-to-Sequence models are typically trained with a pair of input-output examples, $D = \{x_i, y_i\}_{i=1}^{N}$ using a simple token-level cross-entropy (CE) loss. The objective is to maximize the conditional probability of generating the reference or the gold tokens at each timesteps. The token-level losses are aggregated in a batched fashion to update the parameters (θ) of the model. Formally,

$$\mathcal{L}_{AR} = \sum_{(x_i, y_i)} \sum_{t=1}^{T} \log P(y_t | x_i, y_{<t}; \theta)$$

There are two known problems with using CE as a training objective as discussed in Leblond et al. (2017): 1) it introduces a discrepancy between training and inference (exposure-bias) as the model is conditioned on reference-prefix during training and model-generated prefix during inference and 2) it fails to approximate the true global objective, for example, maximizing translation quality as measure by BLEU for MT. Hence, prior work also proposes different loss functions for training these sequence-to-sequence models to account for these limitations (Shen et al., 2016; Edunov et al., 2018; Welleck et al., 2019; Wieting et al., 2019). Nonetheless, the CE loss still remains the most widely used loss function to train encoder-decoder-based models due to its simplicity and efficacy over more complicated loss variants.

2.2.1 Sequence-to-Sequence Models for TS and MT

Both TS and MT, in prior work, have been typically modeled using an autoregressive sequence-to-sequence model that learns to perform the NLG task using a corpus of paired

input-output examples (Graves, 2012; Sutskever et al., 2014; Bahdanau et al., 2015; Vaswani et al., 2017; Specia, 2010; Nisioi et al., 2017; Zhang and Lapata, 2017; Wubben et al., 2012; Scarton and Specia, 2018a; Nishihara et al., 2019; Martin et al., 2020a; Jiang et al., 2020). While both TS and MT are usually framed as sequence generation tasks, they can also be formulated as sequence refinement tasks.

2.2.2 Text Generation as a Sequence Refinement Task

Iterative sequence refinement tasks as a Markov Decision Process is modeled by $(\mathcal{Y}, \mathcal{A}, \mathcal{E}, \mathcal{R}, \boldsymbol{y}^0)$. A state $y = (y_1, y_2, ..., y_L) \in \mathcal{Y}$ is a sequence of tokens where each y_i represents a token from the vocabulary $\mathcal{V}$, L is the sequence length and $y^0 \in \mathcal{Y}$ is the initial sequence to be refined, using actions drawn from the set $\mathcal{A}$. The reward $\mathcal{R}$ is based on the distance $\mathcal{D}$ between the generated output and the reference sequence $y^* \in \mathcal{Y}$: $\mathcal{R}(y) = -\mathcal{D}(y, y^*)$. At each decoding iteration, the model takes an input y, chooses an action $a \in \mathcal{A}$ to refine the sequence using a policy π, resulting in state $\mathcal{E}(y, a)$. Models differ based on the nature of edit actions used and support different operations such as insertion, deletion, reposition, and substitution. In this framework, $P(Y|X)$ at a refinement step t is given by $P(Y_t|X, Y_{t-1})$, and typically modeled via non-autoregressive (NAR) models as they assume conditional independence in the output token space.

These NAR edit-based models have primarily been used to speed up MT by allowing parallel edit operations on the output sequence (Lee et al., 2018; Gu et al., 2018; Ghazvininejad et al., 2019; Stern et al., 2019; Chan et al., 2020; Xu and Carpuat, 2021). Refinement approaches have been used to incorporate terminology constraints in machine translation, including hard (Susanto et al., 2020) and soft constraints (Xu and Carpuat, 2021). They have also shown promise for Automatic Post Editing (APE) (Gu et al., 2019; Wan et al.,

	OPERATIONS	Roll-In	ROLL-IN POLICIES
Gu et al. (2019)	Insertion, Deletion	Mixed	$y' = \{\mathcal{E}(y^*, \tilde{d}), \tilde{d} \sim \pi_{rnd}\}$ $y_{ins} = \{y' \text{ if } u < \alpha \text{ else } \mathcal{E}(y^s, d^*), d^* \sim \pi^*_{del}\}$ $y_{del} = \{y^s \text{ if } u < \beta \text{ else } \mathcal{E}(\mathcal{E}(y_{ins}, p^*), \tilde{t}), p^* \sim \pi^*_{plh}, \tilde{t} \sim \pi_{ins}\}$
Stern et al. (2019)	Insertion	Expert	$y_{ins} = \{\mathcal{E}(y^*, \tilde{d}), \tilde{d} \sim \pi_{rnd}\}$
Ghazvininejad et al. (2019)	Substitution	Expert	$y_{sub} = \{\mathcal{E}(y^*, \tilde{m}), \tilde{m} \sim \pi_{mask}\}$
Saharia et al. (2020)	Substitution	(Offline) Learned	$y_{sub} = \{\mathcal{E}(y, \tilde{m}), \tilde{m} \sim \pi_{mask}\}$
Qian et al. (2020)	Substitution	Expert	$y_{sub} = \{\mathcal{E}(y, \tilde{m}), \tilde{m} \sim \pi_{mask}\}$
Xu and Carpuat (2021)	Insertion, Reposition (including deletions)	Learned	$y' = \{\mathcal{E}(\mathcal{E}(y^*, \tilde{d}), \tilde{p}), \tilde{d} \sim \pi_{rnd}, \tilde{p} \sim \pi_{per}\}$ $y_{ins} = \{y' \text{ if } u < \alpha \text{ else } \mathcal{E}(y, r), r \sim \pi_{rps}\}$ $y_{rps} = \{y' \text{ if } u < \beta \text{ else } \mathcal{E}(\mathcal{E}(y, p^*), \tilde{t}), p^* \sim \pi^*_{plh}, \tilde{t} \sim \pi_{ins}\}$

Table 2.3: Training Policies and Edit Operations performed by different NAR models: y^s: original input sequence, y^*: output sequence, y: model generated variant of reference sequence, π_{rnd}/ π_{masks} drops/masks random words from y^* according to a distribution (e.g. uniform, bernoulli, etc.), π_p generates a permutation, $u :\sim Uniform[0, 1]$, $\pi_{ins}, \pi_{plh}, \pi_{del}, \pi_{rps}$ are insertion, placeholder prediction, deletion and reposition policies.

2020), and grammatical error correction (Awasthi et al., 2019). In our work, we show that they are a good fit for controllable TS.

Training Objective NAR edit-based models are typically trained via imitation learning that uses a roll-in policy and a roll-out policy. The roll-in policy is used to generate the sequences that the model learns to refine from. A roll-out policy is then used to estimate the cost-to-go from the generated roll-in sequences to the desired output sequences. The cost-to-go is calculated by comparing the model actions to oracle demonstrations. We summarize the policies of various NAR models proposed for MT in Table 2.3.

2.2.3 Edit-based Sequence-to-Sequence Models

Recent work also incorporates edit operations into TS systems more directly. These approaches rely on custom multi-step architectures. They first learn to tag the source token representing the type of edit operations to be performed (e.g. keep, delete, replace) and then use a secondary model for in-filling new tokens or executing the edit operation. The tagging and editing model are either trained independently (Alva-Manchego et al., 2017; Malmi

et al., 2019; Kumar et al., 2020; Mallinson et al., 2020) or jointly (Dong et al., 2019). By contrast, we propose to use a single model trained end-to-end to generate sequences of edit operations to transform the entire source sequence.

2.3 Controllable Natural Language Generation

Controllable NLG, as a conditional language modeling task, can be framed using $P(Y|X, C)$ (Prabhumoye et al., 2020), where X is the source text, and C is the set of attributes we want to control for in the generated text, Y. For example, for MT, C could be the target language and/or the length of the generated output. Controlling attributes in the generated text has numerous applications and is often desirable. For example: in open-ended dialogue generation, controlling the politeness, formality, or topic sequence can enhance the quality of conversation and responsiveness. We first discuss attribute controlled NLG for MT and TS in Section 2.3.1 and then introduce ways to induce control in the models in the section following.

2.3.1 Control in MT and TS

While most machine translation approaches only indirectly capture style properties (e.g., via domain adaptation), our work shares with a growing number of studies that aim to consider source texts and their machine translation in their pragmatic context. Recent attempts at generating multiple translations for a single source have targeted output variability along specific stylistic dimensions like author personality (Mirkin and Meunier, 2015; Michel and Neubig, 2018), politeness (Sennrich et al., 2016a), gender (Rabinovich et al., 2016), formality(Niu et al., 2018a) or to produce diverse translation outputs without a specific use case (Kikuchi et al., 2016; Shu et al., 2019). There have also been efforts at controlling aspects of the simplified output, such as controlling for a specific grade-level (Scarton

and Specia, 2018a; Nishihara et al., 2019) or employing lexical or syntactic constraints (Mallinson and Lapata, 2019; Martin et al., 2020a). Although specifying a desired reading grade level may be easier for non-experts to understand, it offers limited control over the type of simplification. Conversely, exerting control over the simplicity of outputs by adjusting multiple low-level properties allows for more precise control, but it can be burdensome for readers, teachers, or other users to configure. As a result, it is still unclear how to effectively implement and manage the control of text simplification in practical situations. We build upon prior work on controllable TS for grade-specific simplification with an additional constraint of generating output in the desired target language and investigate ways to set low-level control for grade-specific TS.

2.3.2 Inducing Control in Sequence-to-Sequence Models

The techniques used to induce control in natural language texts via language or sequence-to-sequence models can be broadly divided into two categories depending upon whether they modify the training process (via side constraints or discriminative classifiers) or are supplied as constraints during inference.

Training-time Modifications

Control via Side Constraints A straightforward method to capture a target attribute, C, in sequence-to-sequence models is to represent it as a special token (e.g. <2formal>) appended to the input sequence, $[C; X]$, which acts as a side constraint. These constraints can be appended to the source or the target sequence.

The encoder learns a hidden representation for this token as for any other vocabulary token, and the decoder can attend to this representation to guide the generation of the output sequence. This simple strategy has been used in machine translation to control second-

person pronoun forms when translating into German (Sennrich et al., 2016a), formality when translating to French (Niu et al., 2018b), the identity of the target language in multilingual scenarios (Johnson et al., 2016) and to control style, content, and task-specific behavior for conditional language models (Keskar et al., 2019).

However, this approach requires the special tokens that act as side constraints to be added to the vocabulary of the model, and when multiple side constraints are introduced, their order impacts the final output quality. To address this, Schioppa et al. (2021) introduce vector-valued interventions to control multiple attributes simultaneously using a weighted linear combination of the learned vectors for each attribute. The weighted representation is added to the encoder's hidden representations and is accessible by the decoder (see figure to the right). One main advantage of introducing additive intervention over special tags is that it allows for inducing controllability by selectively learning and updating the attribute embedding and the decoder layers of the pre-trained models without training the whole model from scratch. In our work, as we primarily train MT and TS models from scratch on the domain-specific data (with the exception of using unsupervised pre-trained models in Chapter 6), we use the tag-based approach in all our experiments due to its simplicity and efficacy.

Control Using Discriminative Classifiers for Pre-trained Models A second line of work introduces discriminative classifiers or a scoring model to guide the generation process of an already available pre-trained language or sequence-to-sequence model. More formally, given a base model's probability distribution, $P(Y|X)$, a second model is trained to discriminate between good and bad outputs by using a scoring model, $P(C|X,Y)$ (Holtzman et al., 2018). Dathathri et al. (2019) introduce Plug and Play Language Models (PPLM) that uses a discriminator to update the hidden representation of the base model towards attaining a

higher likelihood under a classifier $P(C|Y)$. Li et al. (2022) further improve over PPLMs using Diffusion LMs.

Inference or Decoding-time Constraints

Vocabulary Constraints There are also efforts at controlling target attributes in the text during inference directly without retraining a base generative model. For example, Kajiwara (2019) use complex words as negative constraints when decoding with an autoregressive model for controllable TS. Constrained beam search has also been used for MT to include or remove words or phrases (Hokamp and Liu, 2017). (Ghazvininejad et al., 2017) introduce additional weights corresponding to several features like repetition, word length, sentiment, etc to control the style of a generated poem in the beam search algorithm. Kumar et al. (2021) pose controllable decoding as a multi-objective optimization problem to satisfy multiple constraints (e.g. $P(C|Y)$, $sim(X, Y)$) at the same time. These approaches are nevertheless limited by the generative ability of the base language models.

Prompting Since it is practically infeasible to train or even finetune large-scale language models like GPT-3 (Brown et al., 2020), BLOOM[4] or XGLM (Lin et al., 2021) that are trained on a huge corpus with billions of parameters, more recent work focuses on providing and selecting examples, also known as prompts, that can be used as a context to steer the language model with specific properties. For example, given GPT-3, if the objective is to control the formality of the generated output, the language model, $P(Y|X)$, can be conditioned on the input X:

"Here is some text: Gotta see both sides of the story.

Here is a rewrite of the text that is **more formal**: You have to consider both sides of the story.

[4] https://huggingface.co/docs/transformers/model_doc/bloom

Here is some text: I'd say it is punk though.

Here is a rewrite of the text that is **more formal**:".[5]

The base model then continues the generation of output given the input context, X. Prompting large-scale models have been used to extract knowledge and learned information during pre-training and for several NLP tasks like question answering, machine translation, sentiment classification, and monolingual style transfer among others (Liu et al., 2021b). However, due to a large number of computing resources required to even generate outputs using these models (e.g. BLOOM requires 8x RTX 3090 GPUs) and a lack of transparency on how the model utilizes the context provided vs its own learned information, we do not explore the use of these models in our own work.

2.4 Summary

In this chapter, we provide a detailed overview of the text generation tasks: Machine Translation and Text Simplification, the approaches used to model these tasks, and existing techniques to induce control in sequence-to-sequence models for Controllable Natural Language Generation.

We identify the following gaps in the literature that we aim to close with our work as detailed below:

a) **Multi-Output Machine Translation Generation:** While most prior work focuses on generating a single correct translation, we show that existing sequence-to-sequence models can exhibit diversity in generated translations under desired supervision but fail to capture the full spectrum of the translation distribution as different translations target different properties of the output (Chapter 3).

[5]Example taken from Rao and Tetreault (2018).

b) **Controlling Complexity in Cross-Lingual Settings:** Existing work on controlling the complexity of the output generally targets a simplification in the language of the source text. We introduce the task of Complexity Controlled Machine translation that aims to generate a translated output in a target language for the desired audience group and propose datasets to support the training and evaluation of such models (Chapter 4).

c) **Lack of fine-grained control for target-audience specific simplifications:** While text simplification has been framed as a text editing task in prior work, they usually rely on custom multi-step tag-and-edit architectures and are not yet applied to controllable TS. In our work, we instead adopt general-purpose sequence-to-sequence edit-based NAR models and show that they are a better fit for this task (Chapter 5).

d) **Lack of a systematic method for setting control for target-audience specific simplifications:** Prior work sets low-level control tokens that describe the nature of text simplification operation to be performed by searching for control token values on a development set using an automatic evaluation metric. While appealing in its simplicity, the model is never evaluated for *control* at the instance level as the control token values are always set at the corpus level. We study the appropriate configuration of low-level control tokens to regulate the reading grade level of the text simplification system and introduce data-driven solutions that leverage surface-form features extracted from the source text and the desired target grade level to effectively set these control tokens for grade-specific TS. (Chapter 5).

e) **Inadequate Human Evaluation:** None of the prior work proposes a systematic evaluation of automatic text simplification outputs. We aim to address this gap by performing a thorough human evaluation to directly evaluate the correctness of simplified texts using reading comprehension exercises (Chapter 7).

Chapter 3: Diverse Machine Translation Generation

We start by exploring whether existing MT models can generate the many different valid translations that exist for a given source text. The Duolingo Simultaneous Translation And Paraphrase for Language Education (STAPLE) shared task (Mayhew et al., 2020) provides a framework for developing and testing such systems, grounded in real translations produced by English learners into five native languages (Portuguese, Vietnamese, Hungarian, Japanese, Korean). Given an English sentence prompt, systems are asked to produce a set of translations for that prompt, and are scored based on how well their outputs cover human-curated acceptable translations, weighted by the likelihood that an English learner would respond with each translation (Table 3.1).

Prompt	is my explanation clear?	LRF
Output	minha explicação está clara?	0.267
	minha explicação é clara?	0.161
	a minha explicação está clara?	0.111
	a minha explicação é clara?	0.088
	minha explanação está clara?	0.057
	está clara minha explicação?	0.044
	minha explanação é clara?	0.039

Table 3.1: STAPLE data: given a prompt in English, translation alternatives are weighted according to Learner Response Frequency (LRF).

While the multiple translations can be viewed as paraphrases, we propose to address the STAPLE task primarily as an MT task to better understand the strengths and weaknesses of neural MT architectures for generating multiple learner-relevant translations. Given a

Transformer model for the language pair of interest, we use beam search to generate n-best translation candidates. However, since n-best lists are known to lack diversity, we propose to generate hypotheses that better match the requirements of the STAPLE task via:

a) **Frequency-Aware n-Best Lists**: We encourage hypotheses to reflect the diversity and frequency of learner responses by fine-tuning models on STAPLE data, oversampling translation options to reflect learner preferences.

b) **Hypothesis Filtering**: We filter the resulting n-best lists using a binary classifier which identifies good translations that are likely to be produced by a learner.

Controlled experiments and analysis show the benefits of both strategies: Frequency-Aware n-Best Lists generated by the finetuned MT model better capture the learner's preferences toward different translation options and filtering these n-best lists via a classifier trained to maximize the goodness of the set of the selected translations (LRF weighted F1) further improves performance.

3.1 Approach

3.1.1 Frequency-Aware Hypotheses Generation

While neural MT systems can generate multiple translation candidates per source using beam search, the n-best translations often lack diversity. One issue is that systems are trained on single-translation training samples. We propose to tailor MT to the STAPLE task by fine-tuning models on LRF-weighted multi-reference samples to obtain more diverse translations and a ranking that better reflects learner preferences.

Given the STAPLE data for a language pair, where the i-th training example, $(\mathbf{x}_i, \mathbf{Y}_i, \mathbf{W}_i)$ includes a source sentence in English, a reference set $\mathbf{Y}_i = \{\mathbf{y}_i^1, \mathbf{y}_i^2, ..., \mathbf{y}_i^K\}$ of $\mathbf{K}$ translations and corresponding LRF weights $\mathbf{W}_i = \{\mathbf{w}_i^1, \mathbf{w}_i^2, ..., \mathbf{w}_i^K\}$, we create MT training

samples by copying the translation pair $(\mathbf{x}_i, \mathbf{y}_i^j)$, $(\mathbf{w_i^j} \times O)$ times.[1] Given model parameters θ, this yields a weighted cross-entropy loss:

$$\mathcal{L}_{lrf}(\theta) = \sum_{i=1}^{M} \sum_{j=1}^{K} (w_i^j \times O) \log Pr(y_i^j | x_i; \theta) \tag{3.1}$$

3.1.2 Hypothesis Filtering as Binary Classification

Even when informed by STAPLE data and LRF scores, n-best lists might include translations that are not in the reference set, due to translation errors or selecting paraphrases that do not match language learners' preferences. We design a binary classifier that further filters the n-best lists by predicting for each hypothesis whether or not it should be included in the final set. This lets us define features based on the complete prompt and hypothesis sequence pairs, while the MT model generates the hypothesis incrementally.

Let $D = \{(\mathbf{x}_i, \hat{\mathbf{y}}_i^1, \hat{\mathbf{y}}_i^2, ..., \hat{\mathbf{y}}_i^N)\}_1^M$ represent the n-best list generated via beam search for all the source prompts in the training dataset: $\mathbf{x}_i$ corresponds to the i-th source prompt and $\hat{\mathbf{y}}_i^j$ corresponds to the j-th candidate hypothesis extracted via beam search. $\mathbf{f}_j^i$ represents the feature vector extracted from the source ($\mathbf{x_i}$) and j-th candidate hypothesis ($\hat{\mathbf{y}}_\mathbf{i}^\mathbf{j}$) and $\mathbf{l}_i^j$ is a binary label indicative of whether the candidate hypothesis, $\hat{\mathbf{y}}_i^j$, is found in the gold standard data. The classification model $g : F \rightarrow R$ maps the feature vector to a real value, where, g is a two-layer Neural Network (NN) to enable learning feature combinations.

Features We aim to capture the quality of a source-hypothesis pair using multiple sentence-level features:

- Length features $|\hat{y}|$, $|x|$, $\frac{|\hat{y}|}{|x|}$, $\frac{|x|}{|\hat{y}|}$ might indicate mismatches between source and target content.

[1] We set $O = 1000$ in practice.

- Word alignment features have proved useful to identify semantic divergences in bitext (Munteanu and Marcu, 2005; Vyas et al., 2018). We use the Forward and Reverse Alignment score, the count of unaligned words for source and target, and the top three largest fertilities for source and target.

- Scores from various MT models as often done when reranking n-best lists (Cherry and Foster, 2012; Neubig et al., 2015; Hassan et al., 2018) including a left-to-right model, a right-to-left model, and a target-to-source model, which provide different views of the example and might better estimate the adequacy of the translation than the original MT model score.

- Target 5-gram language model score to estimate the fluency of the hypothesis.

Loss We optimize a Soft Macro-F1 objective (Hsieh et al., 2018) function to approximate the official evaluation metric.[2] The true positive (tp), false positive(fp), and true negative (tn) rate for each source prompt e_i are estimated as:

$$\mathrm{tp}_{\mathrm{x_i}} = \sum_{t=1}^{N} \hat{l}_i \times l_i$$
$$\mathrm{fp}_{\mathrm{x_i}} = \sum_{t=1}^{N} \hat{l}_i \times (1 - l_i)$$
$$\mathrm{tn}_{\mathrm{x_i}} = \sum_{t=1}^{N} (1 - \hat{l}_i) \times l_i$$

[2]Preliminary experiments showed that a LRF-weighted version of this loss resulted in unstable training and inconsistent results depending on initialization.

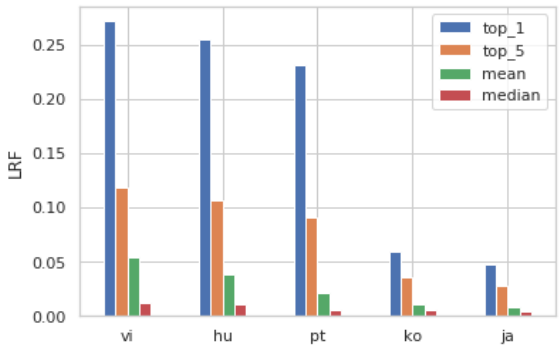


Figure 3.1: Average of the top-1, top-5, mean and median LRF values across source prompts: the LRF distribution is more uniform for languages with many more references per prompt (e.g. en-ja).

Then, the precision, recall, F1 for a source x_i, and the loss are defined as:

$$\mathrm{P}_{\mathrm{x_i}} = \frac{\mathrm{tp}_{\mathrm{x_i}}}{\mathrm{tp}_{\mathrm{x_i}} + \mathrm{fp}_{\mathrm{x_i}} + \epsilon}$$

$$\mathrm{R}_{\mathrm{x_i}} = \frac{\mathrm{tp}_{\mathrm{x_i}}}{\mathrm{tp}_{\mathrm{x_i}} + \mathrm{fn}_{\mathrm{x_i}} + \epsilon}$$

$$\mathrm{F1}_{\mathrm{Macro}x_i} = \frac{2 \times \mathrm{P}_{\mathrm{x_i}} \times \mathrm{R}_{\mathrm{x_i}}}{\mathrm{P}_{\mathrm{x_i}} + \mathrm{R}_{\mathrm{x_i}} + \epsilon}$$

$$\textsc{Loss} = \sum_{i=1}^{M} (1 - \mathrm{F1}_{\mathrm{Macro}x_i})$$

3.2 Experiment Settings

3.2.1 Data

STAPLE Data The STAPLE dataset provides English source prompts, associated with high-coverage sets of plausible translations in five other languages. These translations are weighted and ranked according to LRF scores indicating which translations are more likely. About 3000 prompts per language are available (see Table 3.2 for details) and the number of reference translations available per prompt vary across languages (mean: 174.2, variance: 116). Figure 3.1 illustrates the differences in LRF distributions across languages:

for languages with many references per prompt (e.g. en-ja, en-ko), the gap between the top-1 and the mean LRF value is small, indicating an almost uniform distribution. Average top-1 LRF scores also vary across languages (e.g en-vi: 0.25, en-ja: 0.05) depending upon the number of references available per prompt. For system development, we divide the STAPLE dataset into train, development and test datasets using 72%, 8%, and 20% of source prompts respectively.

Language	Source					Target					T/S
	Train	Dev	Test	Types	Tokens	Train	Dev	Test	Types	Tokens	
EN-PT	2.8K	300	800	2.3K	3.8M	380K	42K	104K	8.7K	4M	131
EN-VI	2.5K	280	700	2.3K	950K	142K	14K	38K	1.7K	1.3M	56
EN-JA	1.8K	200	500	1.3K	3.8M	600K	65K	166K	4K	6.8M	342
EN-KO	1.8K	200	500	1.3K	3M	500K	57K	137K	17K	2.6M	280
EN-HU	2.8K	320	800	1.5K	1.1M	182K	21K	47K	11K	1M	62

Table 3.2: STAPLE data statistics: segments in our train/dev/test split, overall vocabulary statistics and average translations per source prompt (T/S).

Language	OpenSubtitles	Tatoeba
EN-PT	47.2M	196K
EN-VI	3M	5.3K
EN-JA	1.8M	200K
EN-KO	1.2M	2.7K
EN-HU	34.5M	102K

Table 3.3: Additional bitext used for training and fine-tuning MT models

Other Bitexts We use bitext from OpenSubtitles (Tiedemann, 2012) and Tatoeba (Tiedemann, 2012) as described in Table 3.3. The Tatoeba corpus provides multiple reference translations for some sources (with 2 translation per source on average), but unlike in the STAPLE data, these translations are not weighted by frequency of usage.

Preprocessing All datasets are pre-processed using Moses tools for normalization, tokenization and lowercasing. We further segment tokens into subwords using a joint source-

target Byte Pair Encoding (Sennrich et al., 2016b) model with $32,000$ operations. For Japanese, we use kytea [3] toolkit for word tokenization.

3.2.2 MT configurations

Model Architecture We use the Transformer model implemented in the Sockeye toolkit[4] as a baseline MT system. Both encoder and decoder are 6-layer Transformer models with the model size of $1,024$, feed-forward network size of $4,096$, and 16 attention heads. We adopt label smoothing and weight tying. We tie the output weight matrix with the target embeddings. We use the Adam optimizer with an initial learning rate of 0.0002.

Experimental Conditions We train several models with the above configuration:

- **OpenSubs** a baseline model trained and validated on the OpenSubtitles bitext.
- **Unweighted** builds on the baseline by fine-tuning on multi-reference samples including the Tatoeba bitext and STAPLE train. We create one training sample per source-reference pair, and the resulting samples are not weighted.
- **Frequency-Aware** is fine-tuned as the unweighted model except that STAPLE train is oversampled as described in § 3.1.1.

We generate n-best list of translations for various models by running beam search with a beam size corresponding to the desired n.

3.2.3 Filtering Configurations

Classifier The 2-layer feed-forward NN has 5 hidden units and 2 output units. It is trained with the Adam optimizer with an initial learning rate of 0.001 and runs for 2000 epochs

[3] https://github.com/neubig/kytea
[4] https://github.com/awslabs/sockeye

on the internal dev set. The best model is selected based on internal test set performance. We consider two losses: the soft macro F1 loss which approximates the official evaluation metric from the shared task organizers (§ 3.1.2) and the standard cross-entropy loss as a baseline.

Reranking Baseline We compare our NN-based classifer with a standard MT n-best list reranker trained on the development set. We use the n-best batch MIRA ranker (Cherry and Foster, 2012) included in Moses. A threshold to filter candidates in the reranked list is selected by maximizing the Weighted Macro F1 on the development dataset.

Features We use eflomal[5] trained on the Opensubtitles dataset to obtain word alignment between source and translation hypotheses. The language model is trained with the kenlm (Heafield, 2011) toolkit with default hyper-parameters[6] on the target side of the Opensubtitles and the STAPLE dataset. The Right-to-left and Target-to-source MT models were trained on OpenSubtitles (same configuration as in § 3.2.2).

3.3 Evaluation

We evaluate the lowercased detokenized output of the systems on our test dataset using:

Weighted Macro F1 This is the official scoring metric from the shared task, which quantifies how the set of system outputs covers the human-curated acceptable translations, weighted by the LRF of each translation.[7] It is defined as the harmonic mean of unweighted precision (P) and weighted recall (WR) calculated for each prompt $\mathbf{x_i}$, and averaged over all

[5] https://github.com/robertostling/eflomal
[6] https://github.com/kpu/kenlm
[7] https://sharedtask.duolingo.com/

the prompts in the corpus. Specifically, using the same notation as introduced in § 3.1.1, for each translation $\mathbf{T_i}$ generated by the MT model, we have:

$$\mathrm{WTP}_{\mathrm{e_i}} = \sum_{t \in T_i} \sum_{y_i^j \in Y_i} 1[t == y_i^j] w_i^j$$

$$\mathrm{WFN}_{\mathrm{x_i}} = \sum_{f_i^j \notin T_i} w_i^j$$

$$\mathrm{WR}_{\mathrm{x_i}} = \frac{\mathrm{WTP}_{\mathrm{x_i}}}{\mathrm{WTP}_{\mathrm{x_i}} + \mathrm{WFN}_{\mathrm{x_i}}}$$

The weighted Macro F1 (WMF1) is then given by:

$$\mathrm{WMF1}_{\mathrm{x_i}} = \frac{2 \times \mathrm{P}_{\mathrm{x_i}} \times \mathrm{WR}_{\mathrm{x_i}}}{\mathrm{P}_{\mathrm{x_i}} + \mathrm{WR}_{\mathrm{x_i}}}$$

$$\mathrm{WMF1} = \frac{1}{M} \sum_{i}^{M} \mathrm{WMF1}_{\mathrm{x_i}}$$

BLEU@1 We also report the translation quality of the 1-best neural MT output compared against the highest LRF reference translation using the standard BLEU metric (Papineni et al., 2002b).

3.4 Experiment Results

3.4.1 Impact of Frequency-Aware Fine-Tuning

Table 3.4 summarizes the evaluation of n-best lists obtained with our neural MT systems.

Baselines We confirm that the neural MT configuration is sound by comparing our neural MT baseline to the provided AWS system by the shared task organizers. Our baseline

LANGUAGE	METHOD	BLEU@1	n-best size	P	WR	WMF1
EN-PT	AWS	68.9	1	**86.67**	14.47	21.60
	OpenSubs	70.9	10	49.66	39.18	37.39
	Unweighted	61.5	10	72.69	40.58	46.11
	Frequency-Aware	**76.6**	10	67.31	**44.34**	**47.4**
EN-VI	AWS	61.4	1	**65.09**	13.32	19.57
	OpenSubs	55.2	10	29.10	31.38	25.76
	Unweighted	49.8	10	56.43	42.91	41.00
	Frequency-Aware	**71.9**	10	61.61	**54.37**	**51.87**
EN-JA	AWS	**50.6**	1	**67.68**	2.18	4.01
	OpenSubs	32.7	50	2.94	3.47	2.52
	Unweighted	30.1	50	45.71	21.21	24.88
	Frequency-Aware	42.4	50	47.29	**22.83**	**26.57**
EN-HU	AWS	63.4	1	**83.70**	18.12	27.12
	OpenSubs	**64.4**	10	41.51	42.6	37.83
	Unweighted	26.2	10	47.11	29.7	31.62
	Frequency-Aware	51.4	10	52.22	**41.05**	**41.69**
EN-KO	AWS	27.9	1	**60.68**	2.26	4.11
	OpenSubs	9.2	50	12.53	7.41	7.20
	Unweighted	14.8	50	33.82	18.8	19.78
	Frequency-Aware	**30.2**	50	35.31	**20.92**	**21.94**

Table 3.4: Frequency-Aware systems outperform both OpenSubs and Unweighted models for all languages. The size of the n-best list for each model was selected based on the WMF1 score on the development set.

("OpenSubs") improves the BLEU@1 score by 2 points for en-pt, and remains 6 points lower for en-vi, as can be expected given the smaller size of the OpenSubtitles training set. However, the "OpenSubs" n-best lists improve over the AWS baseline according to the official task metric (WMF1), establishing that this system is a good starting point for fine-tuning.

Fine-Tuning The Frequency-Aware n-best hypotheses consistently yield the best Weighted Recall and Weighted Macro-F1 scores for all languages. The improvement in recall and therefore F1 score is largest for en-ja and en-ko which have larger translation reference sets

(Table 3.2). Frequency-Aware oversampling also improves precision over the Unweighted model for all but one language (en-pt). The impact on the auxiliary BLEU@1 metric is less consistent: the Frequency-Aware system achieves the best BLEU@1 in 3 out of 5 languages, but outperforms the OpenSubs baseline in 4 out of 5. BLEU@1 drops when fine-tuning on all the samples without weighting (Unweighted) which we attribute to the increased uncertainty during training as the model is exposed to many different translations for the same source English text.

Input: We live near the border.	**LRF**
chúng tôi sống gần biên giới.	0.250
chúng tôi sống ở gần biên giới.	0.053
chúng tôi sống gần đường biên giới.	0.267
chúng tôi sống bên cạnh biên giới.	0.018
chúng tôi sống ở cạnh biên giới.	0.013
chúng ta sống gần biên giới.	0.036
chúng tôi sống cạnh biên giới.	0.061
chúng tôi sống ở bên cạnh biên giới.	0.004
chúng tôi sống ở gần đường biên giới.	0.052
chúng ta sống ở gần biên giới.	0.004
Precision: 100%, Weighted Recall: 76%	
Input: My family lives in the south.	**LRF**
gia đình tôi sống ở miền nam.	0.285
gia đình của tôi sống ở miền nam.	0.134
gia đình tôi sống ở phương nam.	0.061
gia đình của tôi sống ở phương nam.	0.019
gia đình tôi sống ở phía nam.	0.208
gia đình của tôi sống ở phía nam.	0.099
nhà của tôi sống ở miền nam.	-
nhà tôi sống ở miền nam.	-
nhà của tôi sống ở phương nam.	-
gia đình tôi sống ở nam.	-
Precision: 60%, Weighted Recall: 81%	

Table 3.5: Frequency-Aware 10-best Vietnamese output for two randomly selected English prompts. LRF values are given for translations found in the reference set.

Overall, these results show the benefits of fine-tuning on task-relevant data and shows that incorporating LRF weights via oversampling improves the ranking of n-best hypotheses. This is further illustrated in Table 3.5, which shows the top 10 Vietnamese translations for two randomly sampled source prompts: the Frequency-Aware n-best list yields Weighted Recall of 81% at a Precision of 60% and 76% at a Precision of 100% for the two source prompts respectively, illustrating that the model generates high-quality candidates that cover reference translations well, but not perfectly.

N-Best List Quality How well do n-best translations cover the space of reference learner translations? Figure 3.2 shows the impact of increasing the decoding beam (and resulting n-best list size) from 10 to 500 for the Frequency-Aware model. For en-pt, while weighted recall increases up to 66%, the drop in precision hurts the weighted F1 score. The oracle F1 score, which represents the Weighted Macro F1 at a Precision of 100%, also increases gradually, reaching a score of 76%. This suggests that the raw n-best lists contain many useful translation candidates but need to be filtered down to better match translations preferred by language learners.

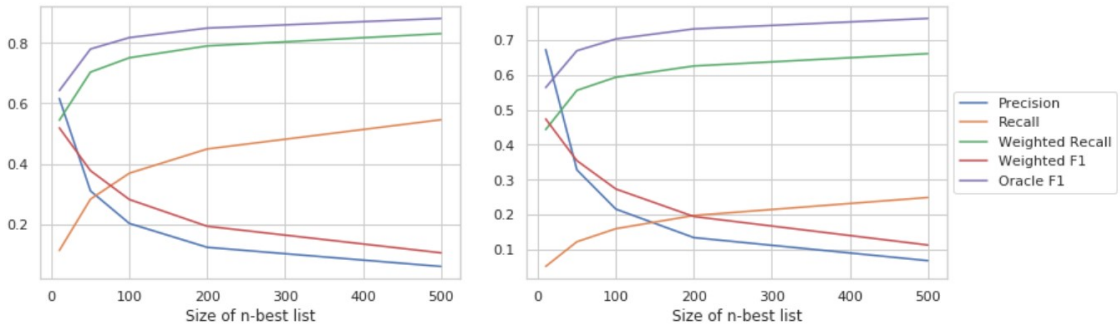


Figure 3.2: Increasing the size of n-best list with the Frequency-Aware system improves the coverage of learner translations for en-pt and en-vi. Oracle F1 is the Weighted Macro F1 at a Precision of 100% and represents the upper bound on WMF1 that can be achieved for a given n-best list.

3.4.2 Impact of Hypothesis Filtering

We explore the impact of hypothesis filtering for en-pt and en-vi:

Filtering consistently improves Precision and Weighted Macro F1 (Table 3.6). The binary classifier that optimizes Soft Macro-F1 performs best, as the loss leads to a better balance between Precision and Weighted Recall than cross-entropy. The classifier outperforms the MIRA reranker. Since the reranker is trained to maximize BLEU@1, it tends to prefer candidates that are lexically similar to the top reference translation and misses some of the more diverse learner translations. This confirms the benefits of framing the selection of candidate hypothesis as binary classification.

METHOD	EN-VI				EN-PT			
	P	**WR**	**WMF1**	**K**	**P**	**WR**	**WMF1**	**K**
No filtering	31.00	**70.31**	37.69	50	44.75	**57.21**	42.84	50
Reranker	**69.71**	46.85	50.67	9	67.44	41.82	45.51	14
Classifier with CE loss	69.70	47.74	48.77	12	69.26	42.70	46.60	10
Classifier with F1 loss	65.15	55.21	**53.69**	12	67.81	45.71	**48.68**	13
Oracle	100	70.31	77.90	15	100	70.31	66.9	16

Table 3.6: Filtering n-best lists consistently improves WMF1 and substantially reduces the size of the output set (**K**).

Ablation Experiments show that the MT scores are the most useful of the features used, as they capture not only the generation probability of a candidate hypothesis but estimate adequacy via the Target-to-source neural MT model (Table 3.7). Length features help precision but not recall, while the alignment and language model scores have little impact overall. This suggests that the classifier could benefit from improved feature design and selection in future work.

FEATURES	P	R	WR	WMF1
All	63.71	15.97	55.91	**54.04**
- LM score	65.10	15.35	55.62	53.92
- Alignment	**65.21**	15.27	55.08	53.86
- Length	58.44	**16.49**	**55.98**	52.53
- MT Scores	43.77	10.88	31.02	28.06
Oracle	100	28.28	70.31	77.90

Table 3.7: Impact of dropping one feature type (§ 3.1.2) at a time from the "All" configuration for en-vi classifier.

3.4.3 Analysis of Translation Diversity

How diverse are the translations returned by various system configurations? Following Zhang et al. (2018b), we quantify diversity using the entropy of k-gram distributions within a translation set:

$$\text{Ent-k (V)} = -\frac{1}{\sum_w F(w)} \sum_{w \in V} F(w) \log \frac{F(w)}{\sum_w F(w)}$$

where V is the set of all k-grams that appear in the translation set, and $F(w)$ denotes the frequency of w in the translations. The higher the Ent-k score, the greater the diversity.

TRANSLATIONS	EN-PT		EN-VI	
	ENT-4	n	ENT-4	n
OpenSubs	2.34	10	2.53	10
Unweighted	2.60	10	2.65	10
Frequency-Aware	2.59	10	2.67	10
Filtered	**2.95**	13	**2.71**	11
Reference	3.93	131	3.23	56

Table 3.8: Diversity in translation sets: Filtered sets are more diverse, bridging 40% of the gap between baseline and reference translations for en-pt.

METHOD	EN-PT				EN-VI			
	P	**R**	**WR**	**WMF1**	**P**	**R**	**WR**	**WMF1**
Unweighted (10-best)	**72.69**	5.53	40.58	46.11	56.43	10.32	42.19	41.00
Unweighted (filtered 50-best)	67.81	9.68	45.71	48.17	63.14	15.23	54.35	51.48
Frequency-Aware (10-best)	67.31	5.07	44.34	47.40	61.61	11.28	54.37	51.87
Frequency-Aware (filtered 50-best)	64.33	6.40	36.94	41.44	**65.15**	15.33	55.21	53.69
C1	65.09	7.13	47.52	49.31	55.04	14.82	57.57	50.73
C2	64.33	7.30	48.67	48.81	60.41	16.07	60.19	**53.57**
C3	66.41	10.32	50.18	50.17	56.19	15.93	54.89	48.05
C4	59.76	**11.60**	**53.56**	**50.79**	53.75	**18.31**	**61.04**	50.78

Table 3.9: Combination of unfiltered 10-best lists (with better precision) and filtered 50-best lists (with better recall) improves Weighted Macro F1. See § 3.4.4 for details on combinations.

Fine-tuned models improve the diversity of 10-best lists compared to the "OpenSubs" baseline for both en-vi and en-pt (Table 3.8). Overall filtering bridges 40% and 25% of the gap between baseline and reference learner translations for en-pt and en-vi respectively.

3.4.4 System Combinations

A manual examination of translation sets returned by different models suggests that they make complementary errors. We, therefore, consider combining system outputs by taking the union of the set of translations they return. We evaluate the following combinations (Table 3.9):

C1 Frequency-aware (10-best) + Unweighted (10-best)

C2 Frequency-aware (10-best) + Frequency-aware (filtered 50-best)

C3 Unweighted (10-best) + Unweighted (filtered 50-best)

C4 Union of all of the above.

For en-pt and en-vi, it helps to combine higher precision unfiltered 10-best lists, and higher recall filtered 50-best lists. For en-pt, the union of all outputs (C4) performs best

overall. Recall increases when combining the Frequency-Aware and the Unweighted model (C1) compared to individual lists (Unweighted: +1.6, Frequency-Aware: +2) without compromising Precision. Similar trends are observed when adding the filtered 50-best list to unfiltered 10-best lists (C2: +2.2, C3: +4.8). For en-vi, a different combination (C2) yields the best result, perhaps due to the smaller set of reference translations per source prompt (en-vi: 56, en-pt: 131) and high Precision of the "Unweighted" model for en-pt.

3.5 Official Results

We tested our systems on the official blind development set to select the best-performing models for final evaluation on the test set. For Portuguese and Vietnamese, our official submissions include frequency-aware hypothesis generation and hypothesis filtering:

en-vi C2: Frequency-aware (10-best) + Frequency-aware (filtered 50-best)

en-pt C4: Frequency-aware (10-best) + Frequency-aware (filtered 50-best) + Unweighted (10-best) + Unweighted (filtered 50-best)

We did not build hypothesis filtering models for the other languages, and submitted systems based only on unfiltered models:

en-ja Frequency-aware (50-best) + Unweighted (50-best)

en-hu Frequency-aware (10-best) + Unweighted (10-best)

en-ko Frequency-aware (50-best) + Unweighted (50-best)

Table 3.10 and 3.11 compares our submissions to baselines, as well as top and median submissions across participants, for all the languages. On our focus languages (en-pt and en-vi), where systems benefitted from both frequency-aware generation and filtering models,

METHOD	EN-VI	EN-PT	EN-JA	EN-HU	EN-KO
AWS	0.210	0.211	0.042	0.298	0.040
Fairseq	0.267	0.151	0.031	0.130	0.054
Median	0.382	0.451	0.214	0.298	0.047
Top	0.547	0.557	0.316	0.598	0.413
Ours	0.537	0.538	0.283	0.492	0.254

Table 3.10: Excerpt from official results: weighted Macro F1 on the STAPLE dev set

METHOD	EN-VI	EN-PT	EN-JA	EN-HU	EN-KO
AWS	0.198	0.213	0.043	0.281	0.041
Fairseq	0.254	0.136	0.033	0.124	0.049
Median	0.377	0.436	0.239	0.452	0.230
Top	0.558	0.552	0.318	0.555	0.404
Ours	0.539	0.525	0.294	0.469	0.255
Rank	2nd	2nd	2nd	2nd	3rd

Table 3.11: Excerpt from official results: weighted Macro F1 on the STAPLE test set

our submissions obtain a Weighted Macro F1 score of 0.539 for en-vi and 0.525 for en-pt on the official test set, achieving a rank of 2nd on the leader-board, within 2% of the top performing submission. On the other language pairs, where our submissions did not use any filtering, Weighted Macro F1 outperform the baselines and median submission consistently. Interestingly on the en-ja and en-hu task, our system ranks second amongst all the submissions despite not using any filtering. The use of the Tatoeba corpus in the submitted systems (also used in our model submissions) was shown to have the most significant positive impact on metrics (Mayhew et al., 2020). An alternative top-scoring system (Li et al., 2020) also used voting-based beam filtering on translations generated from an ensemble, where only sentences attested by multiple subsystems are retained which resulted in a relatively higher scoring system. As our goal was to study the extent to which a single NMT model can capture the diverse spectrum of human translations, we did not explore ensembling techniques. However, this can be an interesting avenue for future work.

3.6 Conclusion

We proposed two strategies to obtain multiple outputs that mimic translations by produced by language learners from a standard neural MT model. Our experiments showed that (1) finetuning MT models using all reference translations and their weight yields more diverse n-best hypotheses that better reflect learner preferences, and (2) filtering these n-best lists using a feature-rich classifier trained to maximize an approximation of the STAPLE evaluation metric yields further improvements.

While these results suggest that some degree of output diversity can be achieved with little change to core neural MT models, oracle scores obtained with unfiltered n-best lists indicate that current MT systems fail to capture the full spectrum of learners' preferences and language proficiency levels reflected by their choice of vocabulary, sentence structure, and the style of the output. Hence, in the next chapter, we propose a task that directly captures the user's reading proficiency level and propose datasets and models tailored toward this task.

Chapter 4: Complexity Controlled Machine Translation

Generating text at the right level of complexity can make machine translation (MT) more useful for a wide range of users. Readers would also benefit from MT output that is better targeted to their needs by being easier to read than the original. Building a model for this task ideally requires rich annotation for evaluation and supervised training that is not available in bilingual parallel corpora typically used in MT. Controlling the complexity of Spanish-English translation ideally requires examples of Spanish sentences paired with several English translations that span a range of complexity levels. We collect such a dataset of English-Spanish segment pairs from the Newsela website, which provides professionally edited simplifications and translations. By contrast with MT parallel corpora, the English and Spanish translations at different grade levels are only comparable. They differ in length and sentence structure, reflecting complex syntactic and lexical simplification operations.

We adopt a multi-task approach to control complexity in neural MT and evaluate it on complexity-controlled Spanish-English translation. Our extensive empirical study shows that multitask models produce better and simpler translations than pipelines of independent translation and simplification models. We then analyze the strengths and weaknesses of multitask models, focusing on the degree to which they match the target complexity, and the impact of training data types and reading grade-level annotation.[1]

[1]Researchers can request the bilingual Newsela data at https://Newsela.com/data/. Scripts to replicate our model configurations and our cross-lingual segment aligner are available at https://github.com/sweta20/ComplexityControlledMT.

4.1 The Newsela Cross-Lingual Simplification Dataset

We build on prior work that used the Newsela dataset for English or Spanish text simplification by automatically aligning English and Spanish segments of different complexity to enable complexity-controlled machine translation.

The Newsela website provides high quality data to study text simplification. Xu et al. (2015) argue that text simplification research should be grounded in texts that are simplified by professional editors for specific target audiences, rather than more general-purpose crowd-sourced simplifications such as those available on Wikipedia. They show that Wikipedia is prone to sentence alignment errors, contains a non-negligible amount of inadequate simplifications, and does not generalize well to other text genres. By contrast, Newsela is an instructional content platform meant to help teachers prepare curriculum that match the language skills required at each grade level. The Newsela corpus consists of English articles in their original form, 4 or 5 different versions rewritten by professionals to suit different grade levels as well as optional translations of original and/or simplified English articles into Spanish resulting in 23,130 English and 5,320 Spanish articles with grade annotations respectively.

This section introduces our method to align English and Spanish segments across complexity levels, and the resulting bilingual dataset.

4.1.1 Cross-Lingual Segment Alignment

Extracting training examples from this corpus requires aligning segments within documents. This is challenging because text is neither simplified nor translated sentence by sentence, and as a result, equivalent content might move from one sentence to the next. Past work has introduced techniques to align segments of different complexity within documents of the same language (Xu et al., 2015; Paetzold et al., 2017; Štajner et al., 2018).

Complexity controlled MT requires aligning segments of different complexity in English and Spanish. Existing methods for aligning sentences in English and Spanish parallel corpora are not well suited to this task. For instance, the Gale-Church algorithm (Gale and Church, 1993) assumes that aligned sentences should have similar length. This assumption does not hold if the English article is a simplification of the Spanish article. Consider the following Spanish text and its English translation in Newsela:

Spanish: LA HAYA, Holanda - Te has tomado alguna vez una selfie?, Hoy en día es muy fácil. Solo necesitas un teléfono inteligente.

Google Translated English: THE HAGUE, Netherlands - Have you ever taken a selfie? Today is very easy. You only need a smart phone.

Original English Version: THE HAGUE, Netherlands - All you need is a smartphone to take a selfie. It is that easy.

As a result, we adapt a monolingual text simplification aligner for cross-lingual alignment. MASSAlign (Paetzold et al., 2017) is a Python library designed to align segments of different length within comparable corpora of the same language. It employs a vicinity-driven search approach, based on the assumption that the order in which information appears is roughly constant in simple and complex texts. A similarity matrix is created between the paragraphs/sentences of aligned documents/paragraphs using a standard bag-of-words TF-IDF model. It finds a starting point to begin the search for an alignment path, allowing long-distance alignment skips, capturing 1-N and N-1 alignments. To leverage this alignment flexibility, we apply MASSAlign to English articles and Spanish articles machine translated into English by Google translate.[2] An important property of Google translated articles is that they are aligned 1-1 at the sentence level. This lets us deterministically find the Spanish replacement for the aligned Google translated English version returned by MASSAlign.

[2] https://translate.google.com/

Translation quality is high for this language pair, and even noisy translated articles contain enough signal to construct the similarity matrix required by MASSAlign.

4.1.2 Resulting Dataset

Dataset	# tokens/segment	# sents/segment	# of types	# of tokens
Spanish				
Newsela	50.13	2.17	57,361	7,792,285
Global Voices	22.96	1.03	254,111	15,921,948
News	26.73	1.03	80,840	5,587,307
English				
Newsela	43.37	2.65	39,012	7,139,717
Global Voices	21.93	1.06	222,383	15,208,054
News	23.76	1.04	49,589	4,939,085

Table 4.1: Comparisons of the English-Spanish Newsela corpus with machine translation corpora from OPUS drawn from Global Voices and News Commentary.

We thus create: both samples for complexity controlled MT (x, c_y, y) and traditional monolingual text simplification samples $(x, c_{y'}, y')$ that can be used by the multi-task model (Section 4.2). Since the properties of Newsela monolingual simplification samples have been studied thoroughly by Xu et al. (2015), we present key statistics for the cross-lingual simplification examples only. Table 4.1 contrasts Newsela parallel segments with bilingual parallel sentences drawn from the OPUS corpus (Tiedemann, 2009). We use Global Voices and News Commentary from OPUS corpus as it has the most similar domain to the Newsela data. Aligned segments in Newsela are about twice as long as segments in parallel corpora, and contain more than two sentences on each side on average. By contrast, parallel corpora samples align sentences one-to-one on average.

Articles are distributed across reading levels spanning grades 2 to 12 for both English and English-Spanish pairs. Table 4.2 highlights the vocabulary differences among the different grade levels for the Newsela Spanish-English corpus. The vocabulary size of the corpus

corresponding to lower grade level is smaller as compared to higher complexity levels. Also, complex sentences have more words per sentence on average but fewer sentences per segment compared to their simplified counterparts. Simple sentences differ from complex sentences in various ways, ranging from sentence splitting and content deletion to paraphrasing and lexical substitutions, as illustrated in Table 4.3.

GRADE	SOURCE (SPANISH)			TARGET (ENGLISH)		
	TYPES	TOKENS/SEGMENT	SENTS/SEGMENT	TYPES	TOKENS/SEGMENT	SENTS/SEGMENT
2	-	-	-	3749	34.31	3.76
3	2,628	38.59	3.52	10,615	35.57	3.28
4	8,431	39.33	2.95	10,414	39.38	3.09
5	17,082	40.96	2.59	18,508	42.18	2.87
6	16,945	42.78	2.25	16,613	44.21	2.62
7	22,352	46.65	2.19	23,617	47.54	2.57
8	19,317	47.07	1.96	17,746	46.66	2.24
9	24,846	50.08	1.87	22,230	50.08	2.25
10	482	49.19	1.90	341	38.23	1.70
12	42,355	53.98	1.96	-	-	-

Table 4.2: Grade level Statistics of the Newsela Spanish-English corpus. Vocabulary size decreases with the reading grade level. Simpler segments contain fewer sentences and are often shorter that complex segments.

4.2 A Multi-Task Approach to Complexity Controlled MT

Task We define **complexity controlled MT** as a task that takes two inputs: an input language segment x and a target complexity c_y. The goal is to produce a translation $\hat{y}$ in the output language that has complexity c_y. For instance, given input Spanish sentences in Table 4.3, complexity controlled MT aims to produce English translations at a specific level of complexity, which might differ from the complexity of the original Spanish.

Model We model $P(y|x, c_y; \theta)$ as a neural encoder-decoder with attention (Bahdanau et al., 2015). This architecture has been used successfully for the two related tasks of text simplification (Wang et al., 2016; Zhang and Lapata, 2017; Nisioi et al., 2017; Scarton and Specia, 2018a) and machine translation (Bahdanau et al., 2015). The encoder constructs

c_x	SPANISH (x)	c_y	ENGLISH (y)	OPERATION
9	Doug Ratliff, un empresario de 67 años de edad de Richlands, Virginia, dijo que la elección de Trump sería uno de los días más felices de su vida.	3	Doug Ratliff is a businessman from Virginia. Ratliff said Trump's election would be one of the happiest days of his life.	Splitting; Deletion
12	Incluso antes de haber nacido, Daliyah Marie Arana, según dicen sus padres, estaba aprendiendo a leer.	4	Daliyah Marie Arana has been learning to read since before she was born.	Paraphrasing
9	Kes Gray es el escritor de la serie de cuentos de animales "Oi Frog and Friends". A él no le interesaron mucho los descubrimientos del estudio. Los autores y los ilustradores solo necesitan que los personajes principales animales de sus historias sean adorables, concluyó.	5	Kes Gray is the writer of the rhyming animal series "Oi Frog and Friends." He was not bothered by the study's findings. Writers and artists just need to keep the main animal characters in their stories cuddly, he said.	Lexical substitution

Table 4.3: Cross-lingual Newsela examples: the Spanish text x of complexity, or reading grade level, c_x is automatically aligned to English text s_y of c_y. Simplification transformations range from sentence splitting and deletions to paraphrasing and lexical substitution.

hidden representation for each word in the input sequence, while the decoder generates the target sequence, conditioned on hidden source representations.

We hypothesize that training a single encoder-decoder model to perform the two distinct tasks of machine translation and text simplification will yield a model that can perform complexity controlled MT. We adopt the multi-task framework proposed by Johnson et al. (2016) to train multilingual neural MT systems.

Representing target complexity Target complexity c_y can be incorporated in sequence-to-sequence models as a special token appended to the beginning of the input sequence, which acts as a side constraint. The encoder encodes this token in its hidden states as

any other vocabulary token, and the decoder can attend to this representation to guide the generation of the output sequence. This simple strategy has been used in MT to control second person pronoun forms when translating into German (Sennrich et al., 2016a), to indicate the target language in multilingual MT (Johnson et al., 2016), and to obtain formal or informal translations of the same input (Niu et al., 2018a). In monolingual text simplification tasks (Scarton and Specia, 2018a), the reading grade level has been encoded as such a special token.

Training Data and Objectives Fully supervised training would ideally require translation samples with outputs representing different levels of complexity for the same input segment. However, constructing such data at the scale required to train deep neural networks is expensive and unrealistic. Our multi-task training configuration lets us exploit different types of training examples to train shared encoder-decoder parameters θ. We use the following samples/tasks:

- Complexity controlled MT samples (x, c_y, y): These are the closest samples to the task at hand, but are hard to obtain. They are used to defined the complexity-controlled MT loss

$$\mathcal{L}_{CMT} = \sum_{(x,c_y,y)} \log P(y|x, c_y; \theta) \tag{4.1}$$

- MT samples (x, y): These are sentence pairs drawn from parallel corpora. They are available in large quantities for many language pairs (Tiedemann, 2012) and are used to define the MT loss

$$\mathcal{L}_{MT} = \sum_{(x,y)} \log P(y|x; \theta) \tag{4.2}$$

- Text simplification samples in the MT target language $(x, c_{y'}, y')$ where y' is a simplified version of complexity $c_{y'}$ for input x, which are likely to be available in much

smaller quantities than MT samples.

$$\mathcal{L}_{\textit{Simplify}} = \sum_{(x,c_{y'},y')} \log P(y'|x, c_{y'}; \theta) \tag{4.3}$$

The multi-task loss is simply obtained by summing the losses from individual tasks: $\mathcal{L}_{CMT} + \mathcal{L}_{MT} + \mathcal{L}_{Simplify}$. We leave the exploration of weighted multi-task loss for future work.

4.3 Experiment Settings

We evaluate **complexity controlled MT** using a subset of the 150k Spanish-English segment pairs extracted from Newsela as described in Section 4.1. We select Spanish and English segments that have different reading grade levels so that given a Spanish input, the task consists in producing an English translation that is simpler (lower reading grade level) than the Spanish input. The train/development/test split ensures that there is no overlap between articles held out for testing and articles used for training. We refer to the corresponding training examples as **MT+simplify** since it represents the joint task of translation and simplification.

4.3.1 Evaluation Metrics

We evaluate the truecased detokenized output of our models using three automatic evaluation metrics. We estimate translation quality and simplicity via BLEU and SARI respectively. In the cross-lingual setting, we cannot directly compare the Spanish input with English hypotheses and references, therefore we use the baseline machine translation of Spanish into English as a pseudo-source text. The resulting SARI score directly measures the improvement over baseline machine translation. In addition to BLEU and SARI, we also report Pearson's correlation coefficient (PCC) to measure the strength of the linear

relationship between the complexity of our system outputs and the complexity of reference translations. Here we estimate the reading grade level complexity of MT outputs and reference translations using the Automatic Readability Index (ARI)[3] score, which combines evidence from the number of characters per word and number of words per sentence using hand-tuned weights (Senter and Smith, 1967):

$$ARI = 4.71(\frac{chars}{words}) + 0.5(\frac{words}{sents}) - 21.43 \tag{4.4}$$

4.3.2 Training Data

In addition to the Newsela **MT+Simplify** training examples described above, which are of the form (x, c_y, y), we use monolingual English simplification data, bilingual parallel training data and Spanish simplification data.

Newsela Simplification (En and Es) provides training examples of the form $(x, c_{y'}, y')$, where y and y' are in the same language. We refer to this data as **Simplify** data. It is used for training multi-task models and for auxiliary evaluation on English only. Our version of this corpus has 513k English segment pairs extracted using the method by Paetzold and Specia (2016). Similar to Scarton and Specia (2018a), an original article 0 can be aligned to up to four simplified versions: 1,2,3 and 4. Here 4 denotes the least simplified level and 0 represents the most simplified level. The train split consists of 460k instance pairs whereas the development and test sets consist of roughly 20K instances, drawn from the same articles as the MT+preserve and MT+simplify test set. For Spanish, we have 110k segment pairs, which will be used to train the Spanish simplification baseline.

[3] https://github.com/mmautner/readability

Bilingual Parallel Data (Newsela) We also extract parallel Spanish-English segments from Newsela based on aligned segments between Spanish and English articles that have the same reading grade level. We use this dataset to provide in-domain MT training examples which includes roughly 70k instances.

All datasets are pre-processed using Moses tools for normalization, tokenization and true-casing (Koehn et al., 2007). We further segment tokens into subwords using a joint source-target byte pair encoding model with 32,000 operations (Sennrich et al., 2015).

4.3.3 Sequence-to-Sequence Model Configuration

We use the standard encoder-decoder architecture implemented in the Sockeye toolkit (Hieber et al., 2017). Both encoder and decoder have two Long Short Term Memory (LSTM) layers (Bahdanau et al., 2015), hidden states of size 500 and dropout of 0.3 applied to the RNNs of the encoder and decoder which is same as what was used by Scarton and Specia (2018a). We observe that dot product based attention works best in our scenario, perhaps indicating that the task of complexity controlled translation requires mostly local changes that do not lead to long distance reorderings across sentences. We train using the Adam (Kingma and Ba, 2014) optimizer with a batch size of 256 segments and checkpoint the model every 1000 updates. Training stops after 8 checkpoints without improvement of validation perplexity. The vocabulary size is limited to 50000. We decode with a beam size of 5. Grade side-constraints are defined using a distinct special token for each grade level (from 2 to 12). The constraint token corresponds to the grade level of the target instance.

4.3.4 Baseline

We contrast the multi-task system with pipeline based approaches, where translation and simplification are treated as independent consecutive steps. We train a neural MT model

to perform translation from Spanish to English and other neural MT models to perform monolingual text simplification for Spanish and English respectively. In the first pipeline setup, the output from the translation model is fed as input to an English simplification model while in the other, the output from the Spanish simplification model is fed as input to an translation model. As Scarton and Specia (2018a), we simply use grade level tokens as side constraints on English simplification examples to control output complexity.[4]

4.4 Evaluation of Complexity Controlled MT

We compare pipeline and multitask models on the Newsela complexity controlled MT task (Table 4.4). Overall, results show that compared to pipeline models, multitask models produce complexity controlled translations that better match human references according to BLEU. SARI suggests that multitask translations are simpler than baseline translations, and their resulting complexity correlates better with reference grade levels according to PCC.

The two pipeline models use the same MT system, therefore the difference between them comes from text simplification: using English simplification (first pipeline) outperforms Spanish simplification (second pipeline) according to BLEU and PCC, but not SARI. This can be explained by the smaller amount of Spanish simplification training data, which yields a model that generalizes poorly.

The "All tasks" model highlights the strengths of the multi-task approach: combining training samples from many tasks yields improvements over the "Translate and Simplify" multi-task model which is trained on the exact same data as the pipelines. However, even without additional training data, the multitask "Translate and Simpifly" model improves over baselines mainly by simplifying the output more, which suggests that the simplification component of the multitask model benefits from the additional MT training data.

[4] Additional constraints based on simplification operations were also used in that work but did not provide substantial benefits when operations are predicted based on the input.

COMPLEXITY CONTROLLED MT	BLEU	SARI	PCC
Pipeline Baselines			
Translate then Simplify	21.98	30.4	0.436
Simplify then Translate	17.09	37.4	0.275
Multitask Models			
Translate and Simplify	22.51	44.8	0.572
All Tasks	**22.75**	**45.0**	**0.608**

Table 4.4: Compared to pipeline models, multitask models produce complexity controlled translations that better match human references (BLEU), that are simpler (SARI), and whose resulting complexity correlates better with the target grade level (PCC). Pipeline models are trained on Newsela Simplification data and MT parallel data from Newsela and OPUS. "Translate and Simplify" uses the exact same data in a multi-task model. The "All tasks" model uses all data available, including Newsela MT+Simplify examples.

Qualitative analysis suggests that the multi-task model is capable of distinguishing among different grade levels and the simplification operations performed for different grade levels are gradual. Table 4.5 illustrates simplification operations observed for a fixed grade 12 Spanish input into English with target grade levels ranging from 9 to 3. When translating to a nearby grade level, for example, 9, the model roughly translates the entire input. For lower grade levels such as 7 and 5, lexical simplification and sentence splitting are observed.

4.5 Analysis

4.5.1 Output Grade Analysis

We aim to better understand to what degree models simplify the input text: how often does the output complexity exactly matches that of the reference? Does this change depend on the distance between input and output complexity levels? Table 4.6 compiles Adjacency Accuracy scores (Heilman et al., 2008), which represent the percentage of sentences where the system output complexity is within 1 or 2 grades of the reference text. We derive the reading grade levels from ARI (Senter and Smith, 1967) and conduct this analysis for

c_x: 12	Ahora el museo Mauritshuis está por inaugurar una exposición dedicada a los autorretratos del siglo XVII, que destaca las similitudes y diferencias entre las fotos modernas y las obras de arte históricas.
c_y: 9	Now the museum Mauritois is launching an exhibition dedicated to the 18th century authoritations, highlighting the similarities and differences between modern photos and historical artworks.
c_y: 7	The museum is **now set to open** an exhibition dedicated to the 18th century authoritations, highlighting the similarities and differences between modern photos and historical artworks.
c_y: 5	The museum is now set to open an exhibit dedicated to the 18th century. **It highlights** the similarities and differences between modern photos and historical artworks.
c_y: 3	The museum is now set to open an exhibit dedicated to the 18th century. It **shows** the similarities and differences between modern photos and art works.

Table 4.5: Example of multi-task model outputs when translating grade 12 Spanish into increasingly simpler English.

the best pipeline ("Translate then Simplify") and multitask models ("All Tasks"). These adjacency scores are broken down according to the distance between input and target grade levels.

ADJ. LEVEL	MODEL	SOURCE GRADE - TARGET GRADE									
		1	2	3	4	5	6	7	8	9	10
1	Pipeline	**0.593**	**0.629**	0.594	0.556	0.524	0.493	0.472	0.457	0.448	0.444
	Multitask	0.59	0.626	0.594	**0.561**	**0.529**	**0.504**	**0.482**	**0.467**	**0.458**	**0.453**
2	Pipeline	**0.717**	**0.759**	**0.786**	0.747	0.713	0.678	0.654	0.637	0.626	0.621
	Multitask	0.711	0.755	0.784	**0.753**	**0.725**	**0.696**	**0.67**	**0.653**	**0.642**	**0.636**

Table 4.6: Adjacency ARI accuracy within grade level given by Adjacency level for the system output with respect to the target grade: Multitask model is able to better capture the target grade than the pipeline model when the difference between the source and the target grade is greater than 3.

When the source and target grades are close, roughly 60% of system outputs that are within a ± 1 window of the correct grade level. The pipeline model matches the target grade slightly better than the multitask model. However, in the more difficult case where the difference between source and target grades is larger than three, the multitask model

outperforms the pipeline. Increasing the adjacency window to ± 2 pushes the percentage of matches in the 70s.

Overall these results show that multitask and pipeline models are able to translate and simplify, but that they do not yet fully succeed at precisely controlling the complexity of their output to match a specific target reading grade.

4.5.2 Training Data Ablation Experiments

NEWSELA			OPUS	EVALUATION METRICS		
S	MT+S	MT	MT	BLEU	SARI	PCC
✓	✓	✓	✓	22.75	45.0	0.608
	✓	✓	✓	22.62	44.5	0.584
✓		✓	✓	22.51	44.8	0.572
✓			✓	19.16	43.4	0.468
✓	✓	✓		14.65	41.4	0.521

Table 4.7: Data ablation experiments showing the impact of different types of training examples on multi-task model. The OPUS parallel corpus is essential to good performance. Simplification data (S) can be used for simplification supervision when joint translation and simplification examples (MT+S) are unavailable.

Table 4.7 shows the impact of different training data types on the multitask model using ablation experiments. OPUS improves BLEU and SARI performance across the board. However, using OPUS without any Newsela MT data (Row 4) hurts the correlation score, indicating the importance of in-domain MT data to control complexity. The difference between the performance when using joint translation and simplification (MT+S) examples (Row 2) vs. simplification only (S in Row 3) is small in terms of BLEU (+0.11) and PCC (0.012), indicating that the monolingual simplification dataset can provide simplification supervision when MT+Simplify data is unavailable. The overall best performance for the task is obtained by using all types of training examples.

4.5.3 Evaluation on Auxiliary Tasks

In addition to complexity controlled MT, the multi-task model can be used to simplify English text, and to translate from Spanish-to-English without changing the complexity. For completeness, we evaluate on these two auxiliary tasks.

TASK	BLEU	SARI	PCC
English Simplification			
Simplify	55.76	41.7	0.736
Translate and Simplify	**56.47**	41.3	0.730
All Tasks	56.05	**42.1**	0.736
Machine Translation			
Translate	**29.09**	-	**0.769**
Translate and Simplify	27.33	-	0.647
All Tasks	27.63	-	0.658

Table 4.8: Evaluation on auxiliary tasks: Multitask models trained on both the translation and simplification dataset improves the performance for the task of English Simplification.

Table 4.8 summarizes the results: the multitask model slightly outperforms a dedicated simplification model on English simplification, showing the benefits of the additional training data from other tasks. By contrast, on the resource-rich MT task, the standalone translation system performs better. This can be explained by the fact that the standalone system is only responsible for text translation, while the multi-task model is exposed to samples of more diverse complexity levels during training which damage its ability to preserve complexity.

4.5.4 Provenance of Reading Grade Level

Our models control complexity using the gold reading grade level assigned by professional Newsela editors. We investigate the impact of replacing these gold labels with automatic predictions from the ARI metric. ARI can be computed for any English segment, including for MT parallel corpora that are not annotated for complexity.

COMPLEXITY CONTROLLED MT	BLEU	SARI	PCC
Newsela reading grade	22.51	44.80	0.572
ARI on Newsela data	22.26	**45.12**	**0.581**
ARI on all data	20.91	44.75	0.577

Table 4.9: Complexity controlled MT with automatic vs. manual reading grade level tags: ARI provides an adequate substitute for manually assigned grade levels.

Table 4.9 shows that ARI provides an adequate substitute for manually annotated reading grade levels, as BLEU and SARI score remain close when Newsela reading grade levels are replaced by ARI-based tags. However, annotating all data with ARI grades, including the OPUS parallel corpus, hurts BLEU. We attribute this result to the differences in length and number of sentences per segment in OPUS vs. Newsela (Table 4.1): segments of vastly different lengths can have the same ARI score (Equation 4.4), thus confusing the multitask model.

4.6 Conclusion

We introduce a new task that aims to control complexity in machine translation output, as a proxy for producing translations targeted at audiences with different reading proficiency levels. We construct a Spanish-English dataset drawn from the Newsela corpus for training and evaluation and adopt a sequence-to-sequence model trained in a multitask fashion.

We show that the multitask model improves performance over translation and simplification pipelines, according to both machine translation and simplification metrics. The reading grade level of the multi-task outputs correlates better with target grade levels than with pipeline outputs. Analysis shows that these benefits come from their ability to combine larger training data from different tasks. The manual inspection also shows that the multitask model successfully produces different translations for increasingly lower grades given the same Spanish input.

Additionally, we also show that the expensive 3-way parallel data is not necessary to perform this task — multi-task training with only MT and monolingual simplification data is sufficient to approach the performance of the model trained with the cross-lingual simplification data. However, even when simplifying translations, multitask models are not yet able to exactly match the desired complexity level, and the gap between the complexity achieved and the target complexity increases with the amount of simplification required. To close this gap, in the next chapter, we propose models that are able to rewrite and edit texts to achieve better output quality and control over simplified outputs.

Chapter 5: Edit-based NAR Models for Controllable Text Simplification

In controllable text simplification, there is often significant overlap between the source and the target content as shown in the example below: To rewrite the grade 10 original for grade 5, the complex text is split into two sentences and paraphrased. When simplifying for grade 3, phrases are further simplified, and content is entirely deleted. Unlike commonly used autoregressive (AR) models for simplification that learn to copy these operations implicitly, in this work, we argue that edit-based NAR models that iteratively refine an input sequence to reach the desired degree of simplification are a good fit for controllable TS as they have the potential of modeling the simplification process more directly.

GRADE	TEXT
10	Tesla is a maker of electric cars, which do not need gas and can be charged by being plugged into a wall socket.
5	Tesla cars can be charged by being plugged **in, like a phone. They** do not need any gas.
3	Tesla **builds** cars **that** do not need gas.

Table 5.1: Simplified text changes depending on the reading grade level of the target audience. The bold font highlights changes compared to the grade 10 version.

We seek to build on the strengths of general-purpose non-autoregressive edit-based seq2seq models and adapt them for controllable TS. We hypothesize that the key difference between 1) *standard or generic* and *controllable* TS is that in controllable TS, training instances illustrate a more diverse set of relationships between input and output sequences, making it a much more challenging task; 2) MT and *controllable* TS is that the learning

algorithms designed for MT are aligned with inference strategies that generate target from an empty initial sequence. By contrast, in sequence editing tasks like TS, the inference step is initialized instead with a complex source sequence. As a result, standard models require more guidance to make use of these diverse samples effectively.

In this work, we compare and propose strategies to inject more guidance at inference or training time. The inference strategy incorporates lexical complexity information derived from data statistics in the initial sequence to be refined. For training, we introduce a new training framework, EDITING CURRICULUM, which dynamically exposes the model to more relevant edit actions during training and exploits the full spectrum of available training samples more effectively. First, we design a new roll-in strategy, EDITING *roll-in*, that exposes the model to intermediate sequences that it is more likely to encounter during inference. Second, we introduce a training CURRICULUM to expose the model to training samples in order of increasing edit distance, thus gradually increasing the complexity of oracle edit operations that the model learns to imitate.

Our experiments on the Newsela English corpus show that both approaches generate simplified outputs that match the target reading grade level better than strong AR baselines. Our training approach significantly improves the quality and degree of control over the outputs compared to incorporating lexical complexity constraints during inference. Our analysis shows that the sequences generated by our proposed policy enable exploration during training and are easier to learn from, leading to better generalization across samples with varying edit distances. Training with curriculum further improves output quality.

5.1 An Edit-based approach for Controllable TS

Task Definition Given a complex text and a target grade level, the goal of *controllable* TS is to generate a simplified output that matches the desired grade level.

Model We use EDITOR (Xu and Carpuat, 2021), a NAR Transformer model where the decoder layer is used to apply a sequence of edits on an initial input sequence (possibly empty). The edits are of two types: (1) reposition and (2) insertion. The reposition layer predicts the new position of each token (including deletions). The insertion layer has two components: the first layer predicts the number of placeholders to be inserted and the fill-in layer generates the actual target tokens for each placeholder. At each iteration, the model applies a reposition operation followed by insertion to the current input. This is repeated until two consecutive iterations return the same output, or a preset maximum number of operations is reached.

Control tokens The target complexity c_y is encoded as a special token added at the start of the input sequence.

Training EDITOR is trained via imitation learning that uses a roll-in policy to generate the sequences that the model learns to refine from and a roll-out policy to estimate the cost-to-go from the generated roll-in sequences to the desired output sequences. The latter is performed by comparing the model actions to oracle demonstrations where the oracle is the Levenshtein edit distance (Gu et al., 2019). The roll-in sequences for the reposition (or insertion) module are stochastic mixtures of the output

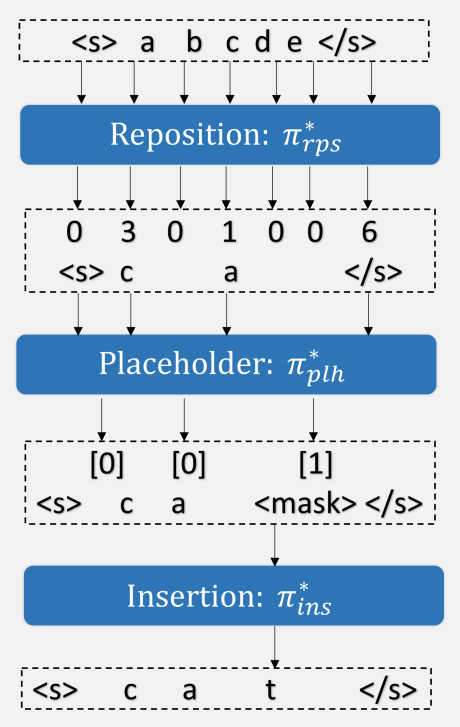


Figure 5.1: One refinement iteration for the input sequence: ”a b c d e” using the operations generated by the Levenshtein Edit Distance Algorithm.

of the insertion(or reposition) module or a noised version of the output sequence. The noisy sequence is generated by applying random word dropping (Gu et al., 2019) and random word shuffle (Lample et al., 2018) with a probability of 0.5 and maximum shuffle distance of 3.

Figure 5.1 shows an example instantiation of the edit actions generated by the Levenshtein Edit Distance to transform the original source sequence ("a b c d e") to the target sequence ("c a t"). In this example, the oracle action is to delete the tokens ["b", "d", "e"], reposition "a" and "c" and insert "t" at the appropriate position. The reposition and the insertion modules are trained in a supervised fashion to predict these oracle operations during training.

We tailor EDITOR for the task of Controllable TS as follows:

5.1.1 Controllable TS Via Lexical Complexity Constraints

We automatically identify the source words that are too complex for the desired target grade and delete them from the initial sequence to be refined by EDITOR (see Figure 5.2. This simple strategy provides finer-grained guidance to the simplification process than the sequence-level side-constraint while leaving the EDITOR model the flexibility to rewrite the output without constraints. We quantify the relatedness between each vocabulary word (w) and grade-level (c) using their Pointwise Mutual Information (PMI) in the newsela corpus (Nishihara et al., 2019; Kajiwara, 2019):

$$PMI(w, c) = log\frac{p(w|c)}{p(w)} \tag{5.1}$$

Here, $p(w|c)$ is the probability that word w appears in sentences of grade level c and $p(w)$ is the probability of word w in the entire training corpus.

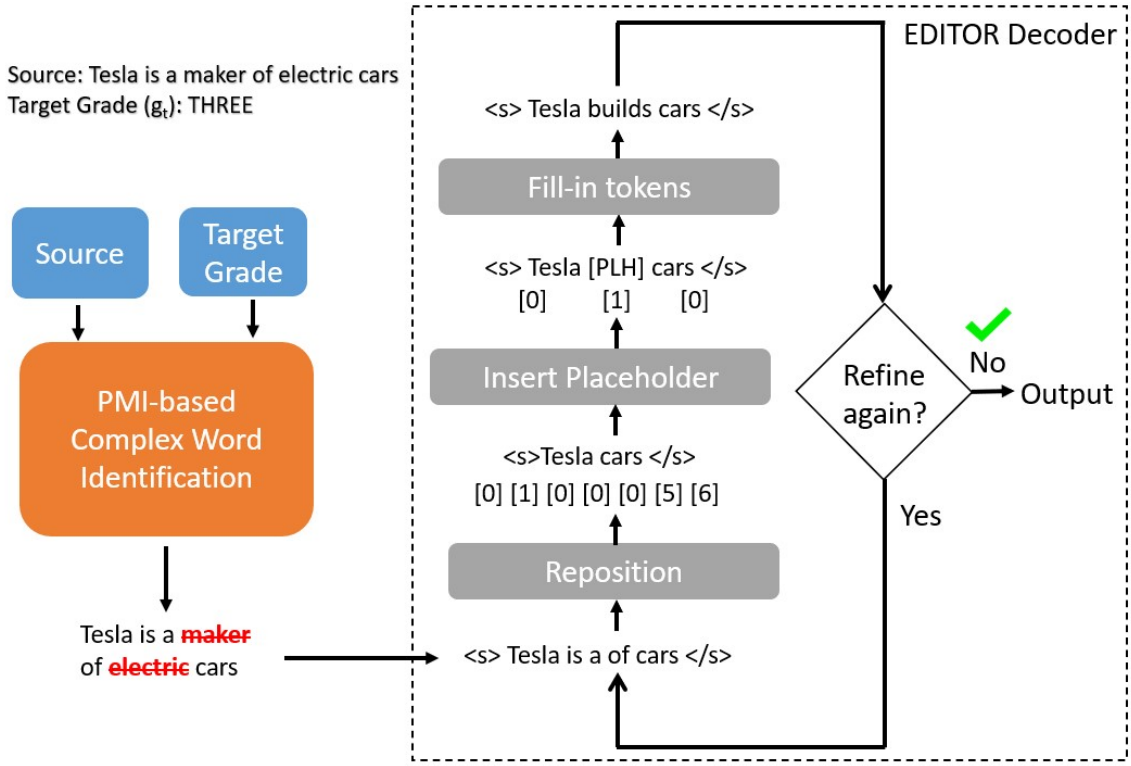


Figure 5.2: EDITOR iteratively refines a version of the input where words predicted to be too complex for 3rd grade readers have been deleted.

While the desired grade level c_y is known in the task, we automatically predict the complexity c_x of each source sentence x using the Automatic Readability Index (4.4). The initial decoding sequence y^s takes the source sequence, x and deletes all words that are strongly related to the source grade level and unlikely to be found in the text of the target grade level, with the exception of named entities:

$$y^s = \{w | w \in x \wedge \ \sim (PMI(w, c_x) > 0 \wedge \ PMI(w, c_y) < 0 \wedge \ w \notin E_x)\} \tag{5.2}$$

where, E_x represents the set of entities in the source sequence x.

Our approach contrasts with prior work where PMI has been used in the loss to reward the generation of target grade-specific words for Controllable TS (Nishihara et al., 2019) or

to exclude complex words from the decoding vocabulary using hard constraints for Generic TS (Kajiwara, 2019). Our approach combines lexical complexity information from both the source and target grade level more flexibly. Starting from y^s as an initial sequence, EDITOR can still delete further content to match the target grade level, insert new words to fix fluency and preserve the original meaning, and has the flexibility to re-generate tokens that were incorrectly dropped from the initial sequence.

Training to generate & refine For Controllable TS, we combine both, training EDITOR to generate text based on the corrupted *target sequence* first, and then fine-tuning the model for refinement based on the corrupted *source sequence* next.

5.1.2 Controllable TS using EDITING CURRICULUM

While injecting lexical complexity signals during inference can provide better control over the generated outputs, we hypothesize that the dependence on those signals is mostly due to lack of a proper exposure to the search space and the training samples during training. We modify the roll-in policy to better match the intermediate sequences encountered at inference and introduce a curriculum to increase the difficulty of oracle actions throughout training.

EDITING Roll-in Sequences generated using the roll-in policy control the search space explored during training. Those sequences should therefore be representative of the intermediate sequences generated at inference time (Ross and Bagnell, 2010). While typically, the roll-in policy is a stochastic mixture of the model and the expert demonstrations as described above, the noise incurred early on due to the large difference between the expert demonstration and the learner's policy actions may hurt overall performance (Brantley et al., 2019; He et al., 2012; Leblond et al., 2018). As we will see (§5.3), this is what happens on

editing tasks when training the model to imitate experts using learned roll-in sequences. At the same time, rolling in with expert demonstrations raises its own issues, as it can limit the exploration of the search space.

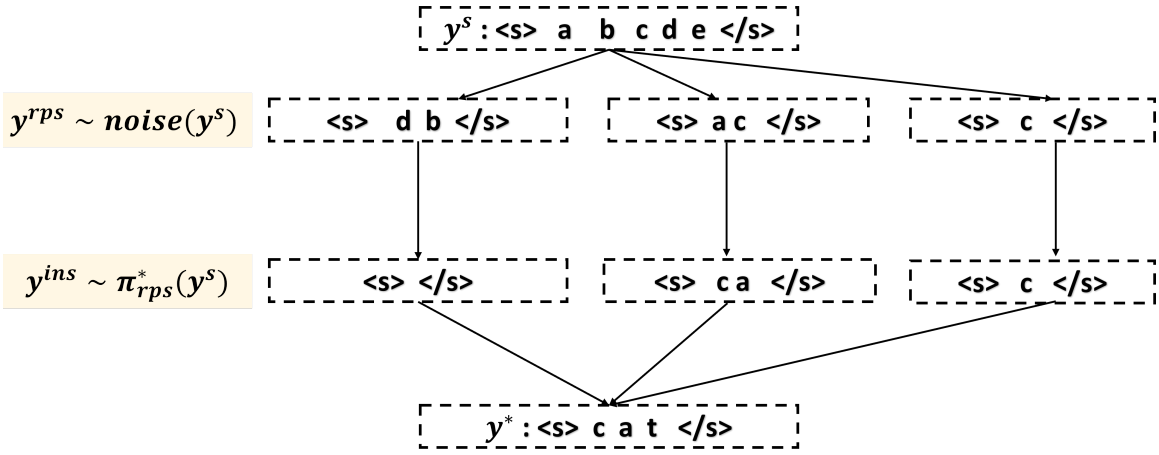


Figure 5.3: Example roll-in sequences for the reposition and the insertion modules: The same initial input sequence (y^s) can enable the model to learn to generate the reference output (y^*) using different edit operations from its noised version.

Motivated by these observations, we propose a new policy, EDITING roll-in, that allows exploration by injecting noise to the input sequence to generate new intermediate sequences for training. This lets the model learn to fix errors without deviating from learning the task at hand. Figure 5.3 shows an example of intermediate sequences generated by our proposed roll-in policy. Different intermediate sequences encourage the model to learn different reposition and insertion edit operations starting from the same input sequence, hence enabling exploration. We modify the roll-in policies to be aligned with the editing inference process, where the reposition operation is followed by an insertion operation on the original input sequence:

- The roll-in sequence for training the reposition module, π_{rps}, is generated by applying noise to the original source sequence y^s, i.e. $y_{rps} = noise(y^s) = \{\mathcal{E}(\mathcal{E}(y^s, \tilde{d}), \tilde{p}), \tilde{d} \sim \pi_{rnd}, \tilde{p} \sim \pi_{per}\}$. Unlike EDITOR, the random word dropping ($\tilde{d} \sim \pi_{rnd}$) and the

word shuffling ($\tilde{p} \sim \pi_{per}$) are applied to the original input sequence instead of the output sequence. This aligns the training with the inference scenario where the model edits an original input sequence instead of generating an output from scratch.

- The roll-in sequence for training the insertion module, π_{ins} is an intermediate sequence generated by applying the expert reposition policy to y_{rps}, i.e. $y_{ins} = \{\mathcal{E}(y_{rps}, r^*), r^* \sim \pi^*_{rps}\}$. The expert reposition policy corresponds to the deletion and reposition actions derived by using the levenshtein edit distance algorithm between the noisy input sequence, $noise(y^s)$ and the target sequence, y^*.

Curriculum controlled roll-out To prevent undertraining when samples with large edit distances overwhelm the loss, we use a curriculum to expose the model to easy-to-learn actions first, then gradually increase the difficulty of the edit-operations performed as the learner becomes more competent. Prior work on curriculum learning (CL) does not agree on standard measures of sample difficulty for seq2seq tasks (Kumar et al., 2019; Yao et al., 2021; Zhang et al., 2018a; Zhou et al., 2020) or apply CL for the different problem of shifting the training of a Transformer model from AR to NAR regimes (Guo et al., 2020; Liu et al., 2021a). By contrast, in our settings, the Levenshtein distance provides a measure of difficulty that directly aligns with the model design and the training oracle.

Resulting Algorithm Given a training dataset $\mathcal{D} = \{y^s, y^*\}_{i=1}^{M}$ consisting of $\mathcal{M}$ samples, the difficulty score $d(s_i)$ for each sample $s_i = \{y^s_i, y^*_i\} \in \mathcal{D}$ is measured by the Levenshtein Distance between the input and the output sequence. The cumulative density function (CDF) of the difficulty scores results in one difficulty CDF score per sample, $\tilde{d}(s_i)$. At each training step t, we estimate the progress made by the learner by computing the competence of the

model $c(t) \in (0, 1]$ as follows:

$$c_{\text{sqrt}}(t) = \min\left(1, \sqrt{t\frac{1-c_0^2}{\lambda_t} + c_0^2}\right)$$

where, λ_t defines the length of the curriculum[1]; $c_0 = 0.1$ as in Platanios et al. (2019).

Based on this competence value $c(t)$, the model is then trained on all the samples whose difficulty as measured by the Levenshtein distance between the input and the output sequence is lower than that competence value, i.e. $\tilde{d}(s_i) \leq c(t)$. The resulting algorithm is also shown in Algorithm 1.

Algorithm 1: Our proposed framework: EDITING CURRICULUM

Input: Dataset, $D = \{y, y^*\}_{i=1}^{M}$, difficulty scoring function, d, and competence function, c.

Compute the difficulty, $d(s_i)$, for each $s_i = \{y_i, y_i^*\} \in D$.
Compute the cumulative density function (CDF) of the difficulty scores. This results in one difficulty CDF score per sample, $\tilde{d}(s_i) \in [0, 1]$
Initialize π_{rps} and π_{ins}.
for *training step* $t = 1...T$ **do**
    Compute the competence value, $c(t)$.
    Create training dataset from by selecting all samples, B_t using $s_i \in D$, such that $\tilde{d}(s_i) \leq c(t)$.
    for *i in* $1..|B_t|$ **do**
        Generate roll-in sequences:
        $y_{rps} = noise(y^s)$
        $y_{ins} = \mathcal{E}(y_{rps}, r^*), r^* \sim \pi_{rps}^*$
        Train π_{rps} and π_{ins} on y_{rps} and y_{ins} minimizing cost-to-go to y^*.
Return **best** π_{rps} and π_{ins} evaluated on validation set.

[1] We set the curriculum length to 5K for our experiments.

5.2 Experiment Settings

5.2.1 Data

We use English Newsela samples extracted in 4.3.2 with 470k/2k/19k for training, development, and test sets respectively, referred to as **Newsela-Grade**. Grade side constraints are defined using a distinct special token for each grade level (from 2 to 12) and are introduced as side constraints for both the source and the target grade levels (Scarton and Specia, 2018a).

5.2.2 Automatic Evaluation Metrics

We evaluate system outputs using: **SARI**; **Pearson's correlation coefficient (PCC)** between the complexity of the system and reference outputs as measured by Automatic Readability Index (ARI) and **ARI-Accuracy**.

5.2.3 Model Configurations

Data Preprocessing We pre-process the dataset using Moses tools for normalization, and truecasing. We further segment tokens into subwords using a joint input-output byte pair encoding model with $32,000$ operations.

Architecture We adopt the base Transformer architecture (Vaswani et al., 2017) with $d_{model} = 512$, $d_{hidden} = 2048$, $n_{heads} = 8$, $n_{layers} = 6$, and $p_{dropout} = 0.1$ for all our models. We add dropout to embeddings (0.1) and label smoothing (0.1). The base EDITOR model is trained using Adam with initial learning rate of 0.0005 and a batch size of $16,000$ tokens. The model is further finetuned on the editing task with a learning rate of 0.0001. We train all our models on two GeForce GTX 1080Ti GPUs. The average training time for a single

seed of AR model is ∼8-9 hrs and for the EDITOR model is ∼20-22 hrs. Fine-tuning EDITOR takes additional 5-6 hrs. Training stops after 8 checkpoints without improvement of validation perplexity. All models are implemented using the Fairseq toolkit. We use ARI to compute the input grade level at the inference time.

Preliminary We establish that our Transformer architecture choice is strong on the more standard generic TS task, as it performs comparably to the state-of-the-art[2] (Jiang et al., 2020) on the Newsela-Auto corpus (Table 5.2).[3]

	SARI	ADD-F1	KEEP-F1	DEL-P
Results as reported in Jiang et al. (2020)				
EditNTS	35.8	2.4	29.4	**75.6**
Transformer-BERT	36.6	4.5	31.0	74.3
AR Transformer (ours)	36.1	3.8	**33.5**	71.1

Table 5.2: Generic TS Evaluation on Newsela-Auto: our Transformer baseline is comparable to SOTA models.

Models We compare our proposed approaches against the following models:

a) **AR** is a auto-regressive (AR) transformer model (Scarton and Specia, 2018a).

b) **AR + PMI-based constraints** is an AR Transformer model which incorporates lexical complexity information as hard constraints during decoding (Kajiwara, 2019): complex words are excluded from beam search using the dynamic beam allocation algorithm (Post and Vilar, 2018). While this approach was introduced for generic TS, we adapt it to controllable TS by defining hard constraints using the same criteria as for deleting words in initial sequences for EDITOR (Section 5.1.1).

[2]The Bert-initialized Transformer has parameters d_{model} = 768, d_{hidden} = 3072, n_{heads} = 12, n_{layers} = 12, and $p_{dropout}$ = 0.1. The encoder and decoder follow the BERT-base architecture. The encoder is initialized with a pre-trained checkpoint and the decoder is randomly initialized.

[3]This corpus contains complex-simple pairs extracted from 1,506 articles for training, 188 for validation and 188 for testing for generic TS.

c) **`AR + PMI weighted loss`** (Nishihara et al., 2019) is an AR Transformer model trained with a loss that weights words based on their PMI values with the desired target grade level.

d) We train EDITOR with the dual-path roll-in policy as in (Xu and Carpuat, 2021), referred to as **`From Reference`**. We fine-tune EDITOR with the following policy variants:

(a) **`From Input`** replaces the reference with the input for generating the initial sequence.

(b) **`From Input + PMI-based initialization`** replaces the source sequence during inference with an initial decoding sequence as detailed in §5.1.1.

(c) **EDITING** is our proposed roll-in policy.

(d) **`Editing Curriculum`**, EDITCL, refers to our approach as described in §5.1.2.

5.3 Automatic Evaluation of Controllable TS

As can be seen in Table 5.3, both our proposed training framework EDITCL and inference adaptation, `From Input + PMI-based initialization` significantly improve over AR baselines. Our inference adaptation approach also outperforms the `AR + PMI-based constraints` baseline across all metrics which oversimplifies the source text by always deleting the complex tokens, as shown by a decrease in keep-F1 (-6.1) and improved del-P (+3.6). By contrast, `From Input + PMI-based initialization` uses lexical complexity information to provide an initial canvas and yields simplified sentences that match the desired target complexity better than the AR baselines. This is reflected in the higher SARI (+14.6) obtained by the `PMI-based initialization` baseline

	MODEL	SARI				ARI-BASED	
		KEEP-F1	ADD-F1	DEL-P	COMBINED	PCC	% ARI-ACC
[1]	Source	67.7	0.0	0.0	22.5	0.632	29.4
[2]	Reference	98.9	88.1	86.3	91.1	1.000	100.0
[3]	PMI-based initialization	60.5	1.3	49.3	37.1	0.614	26.4
[4]	AR	66.2 ±0.3	4.4 ±0.3	43.4 ±1.4	38.0 ±0.5	0.716 ±0.004	34.5 ±0.4
[5]	+ PMI-based constraints	62.2 ±0.7	4.9 ±0.3	47.8 ±0.3	38.3 ±0.1	0.691 ±0.002	35.0 ±0.5
[6]	+ PMI weighted loss	68.2 ±0.3	4.5 ±0.3	42.9 ±1.3	38.5 ±0.5	0.724 ±0.002	36.6 ±0.5
[7]	FROM REFERENCE	66.1 ±0.2	2.2 ±0.2	45.5 ±1.2	37.9 ±0.4	0.656 ±0.003	29.7 ±0.2
[8]	FROM INPUT	66.7 ±0.1	3.1 ±0.1	47.6 ±0.4	39.1 ±0.1	0.731 ±0.001	36.4 ±0.0
[9]	+ PMI-based initialization	66.5 ±0.1	3.5 ±0.1	49.0 ±0.4	39.7 ±0.1	0.737 ±0.000	38.1 ±0.1
[10]	EDITING	66.1 ±0.2	**5.2** ±0.1	51.7 ±0.2	41.0 ±0.0	0.745 ±0.005	39.7 ±0.2
[11]	EDITCL	66.8 ±0.2	4.9 ±0.2	**53.3** ±0.4	**41.7** ±0.3	**0.747** ±0.004	**39.8** ±0.3

Table 5.3: Automatic evaluation results on Newsela-Grade test dataset: our proposed framework, EDITCL, achieves the best performance on SARI and ARI-based metrics across the board.

relative to the `Source`, which represents outputs generated by deleting complex tokens from the source text and hence by itself is not well-formed. `AR + PMI weighted loss` performs comparably to the `AR` baseline across all the metrics except PCC, which could be due to PMI values being a relatively noisy signal at the token level during training, especially for the target grade levels where the data is scarce.[4]

Our overall training framework, `EDITCL` improves over our inference-based strategy significantly across all metrics (i.e., SARI: +2.0, PCC: +0.01, ARI-Acc: +1.7%). Ablation experiments show that this is a combined effect of multiple factors. Dual-path roll-in, `From Input` improves over `From Reference` as expected (SARI: +1.2, PCC: +0.076, ARI-Acc: +6.7%), as the roll-in sequences encountered during training are similar to those encountered during

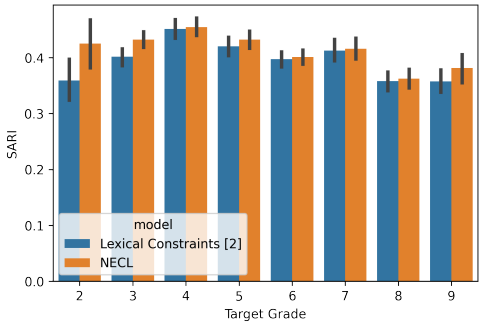


Figure 5.4: SARI score bucketed by the target grade level.

[4]We note that the improvement in SARI reported in prior work (Nishihara et al. (2019): +0.15) is within the confidence interval (+0.5) of the `AR` baseline (Table 5.3).

inference. Using expert roll-in (EDITING) performs better than using learned roll-in (dual-path roll-in) across the board, with gains of up to 3.8 SARI points over `From Reference`. Training with CL (EDITCL) improves over the best roll-in strategy [5], improving the precision of deletions (+1.6) and leading to a significant improvement in SARI score (+0.7) over EDITING with no significant change in grade-specific metrics. Figure 5.4 breaks down simplification quality (SARI) by the desired target grade level.EDITCL achieves the best SARI scores across all the grade levels compared to `From Input + PMI-based initialization .`

5.4 Human Evaluation of Controllable TS

We randomly sample 60 source sentences from the Newsela-Grade dataset, among sources that are simplified toward four distinct grade levels (∼240 examples). For each of these target grades, we obtain ratings of system outputs and references from five Amazon Mechanical Turk workers.[6] Following prior annotation protocols (Jiang et al., 2020), we ask workers to rate outputs on three dimensions: a) is the output grammatical? [0-4] b) to what extent is the meaning expressed in the original sentence preserved in the output? [0-4] and c) how simplified is the output with respect to the original source sentence? [0-10]. Different from prior work, we use a 10-point scale for evaluating simplicity to map the rating resolution to the gold grade differences.

Quality Control We set the location restriction to the United States to control for the quality of annotations. The correlation between the target grade levels and the simplicity ratings of the reference text is 0.582, which suggest that workers do rank simpler output higher than a relatively complex reference of the same source sentence.

[5] As the order of the training samples governed by our curriculum strategy will be the same for `From Input`, EDITING, we only report results over the best roll-in strategy.

[6] We compensate the Amazon Mechanical Turk workers at a rate of $0.03 per HIT.

5.4.1 Task Instructions

We provide the following instructions to the Amazon Mechanical Turk workers to evaluate generated simplified sentences.

Meaning You are given one sentence and 3 rewrites of the same sentence. Carefully read the instructions provided and then use the sliders to indicate the extent to which the meaning expressed in the original sentence is preserved in the rewrites (Agirre et al., 2016).

SCORE	CATEGORY
4	they convey the same key idea
3	they convey the same key idea but differ in some unimportant details
2	they share some ideas but differ in important details
1	they convey different ideas on the same topic
0	Completely different from the first sentence

Grammar You are given three sentences. Carefully read the instructions provided and then use the sliders to indicate the extent to which each of the sentence is grammatical (Heilman et al., 2014).

SCORE	CATEGORY
4	Perfect: The sentence is native-sounding.
3	Comprehensible: The sentence may contain one or more minor grammatical errors
2	Somewhat Comprehensible: The sentence may contain one or more serious grammatical errors,
1	Incomprehensible: The sentence contains so many errors that it would be difficult to correct
0	Other/Incomplete This sentence is incomplete

Simplicity You are given one sentence and 3 rewrites of the same sentence. Carefully read the instructions provided and use the sliders to indicate how simple is each of the rewrite as compared to the original sentence (0: not simplified at all, 10: most simplified). We provide the following examples for your reference.

Original sentence: Craig and April Likhite drove to Chicago from Evanston with their 10-year-old son, Cade, because they wanted to see history made with other fans as close to Wrigley Field as possible.

Rewrites	**Score**
1. Craig and April Likhite drove to Chicago. They wanted to see history made with other fans as close to Wrigley Field as possible. *Explanation:* long sentence has been split into two and complex words are dropped	7
2. Craig and April Likhite to Chicago with their son Cade. *Explanation:* drastic content deletion	8
3. Craig and April Likhite drove to Chicago from Evanston with their 10-year-old son, Cade. They wanted to see history made with other fans as close to Wrigley Field as possible. *Explanation:* long sentence is split into two simple sentences	4
4. Craig and April Likhite drove to Chicago because they wanted to see history made with other fans as close to Wrigley Field as possible. *Explanation:* paraphrasing and deletion	6

Table 5.4: Example illustrating ratings for simplified rewrites of an originally complex sentence.

5.4.2 Findings

We compute the absolute difference ("AbsDiff") in the simplicity ratings between the reference and the system output by the same annotator and aggregate over all examples and all ratings. Table 5.5 shows that our outputs are closer to the reference according to the simplicity judgments than the AR system outputs. The "Mean" ratings indicate that the two models make different trade-offs: where the AR model under-simplifies the source sentence and preserves the meaning, `From Input + PMI-based initialization` almost matches the mean simplicity of the reference at the cost of lower meaning preservation. Our outputs are also less grammatical than those of the AR model and the references, probably due to the independence assumptions made by the non-autoregressive model. The

Adjacency Accuracy, representing the percentage of system outputs within a difference of one rating with the reference, is also higher for From Input + PMI-based initialization relative to the AR model.

	MEANING	GRAMMAR	SIMPLICITY		
	Mean	Mean	Mean	Abs. Diff ↓	Adj. Acc ↑
Reference	2.763	**3.193**	5.325	-	-
AR	**2.803**	3.171	5.157	2.035	0.533
From Input + PMI-based initialization	2.647	3.081	5.310	**1.895**	**0.575**

Table 5.5: Human Evaluation Results: From Input + PMI-based initialization generates output that matches the reference judgments better than the AR baseline.

5.5 Ablation Analysis

5.5.1 Impact of PMI-based Initialization for Inference-Adapted EDITOR

Table 5.6 summarizes the impact of the design choices described in Section 5.1.1: Removing lexical information (`-PMI-based Initialization`) hurts both SARI and the grade-specific metrics. Further, using the baseline EDITOR model that is trained only to generate, without `fine-tuning` for refinement, significantly hurts the performance across the board. In that setting, EDITOR never learns to delete tokens from the source, but only learns to delete tokens inserted by the model. Using EDITOR to generate the output from scratch instead (`-Src Initialization`) recovers the performance on SARI and grade-specific metrics but fails to match the performance of our proposed approach. This shows that fine-tuning for refinement and providing initial sequences informed by lexical complexity are both key to the performance of the EDITOR for Controllable TS.

We also compare From Input + PMI-based initialization with the variant of the model that is trained to perform the reposition operation independent of the insertion operation (`-Joint`), similar to Mallinson et al. (2020). Even though this variant is able to match

MODEL	SARI ↑	%PCC ↑	%ARI-Acc ↑	MSE ↓	ITERATION
From Input + PMI-based initialization	40.2 ±0.0	73.9 ±0.3	36.1 ±0.2	3.73 ±0.03	2.32 ±0.04
−PMI-based Initialization	39.2 ±0.1	73.1 ±0.9	34.5 ±0.4	4.01 ±0.06	2.23 ±0.06
−Finetune	37.6 ±0.4	63.7 ±0.5	28.0 ±0.1	5.12 ±0.01	1.14 ±0.00
−Src Initialization	37.2 ±0.2	72.2 ±0.3	34.4 ±0.4	3.59 ±0.09	2.13 ±0.07
−Joint	39.7 ±0.1	68.1 ±1.7	31.6 ±0.2	4.70 ±0.05	1.00 ±0.00
Single Iteration	40.3 ±0.2	72.6 ±0.1	35.7 ±0.5	3.99 ±0.01	1.00 ±0.00
Gold Source Grade	40.2 ±0.1	74.1 ±0.5	36.5 ±0.3	3.71 ±0.04	2.33 ±0.03

Table 5.6: Ablation analysis for Inference-adapted EDITOR using PMI-based Initialization on Newsela-Grade development set.

SARI, the difference in grade-specific metrics is significant, showing the benefits of joint training of the insertion and reposition components.

METHOD	PRECISION	RECALL
Target Only	76.4	80.2
Source (ARI) Only	**76.6**	78.2
Source (ARI) + Target	76.4	**91.3**

Table 5.7: Using both source and target grade to filter complex words yields maximum overlap with the set of tokens that are preserved from the source in the reference on the Newsela-Grade development set.

Finally, we verify that using ARI to estimate the complexity of the source is effective. Replacing the ARI predictions with the gold-standard grade-level improves the grade-specific metrics only marginally. Table 5.7 further shows the advantage of combining source and target grade information when identifying complex tokens (Equation 5.2) over using source or target grade only.

5.5.2 Impact of EDITING Roll-in

We seek to measure whether our approach has the intended effect of bridging the gap between training and test for editing tasks. Figure 5.5 shows the distribution of oracle insertion and deletions observed when (a) training with EDITOR ś default roll-in policy;

(b) refining an original input sequence and (c) exposed to the model with our EDITING roll-in policy for Controllable TS. The plots show that with the default learning policy of the Editor model, the model doesn't learn to perform complex deletion operations at inference time. By contrast, our proposed roll-in exposes the model to the distribution that has higher overlap with the inference distribution as well as additional intermediate sequences that encourages exploration during training.

5.5.3 Impact of Curriculum Controlled Roll-out

Training Dynamics To verify that curriculum learning helps our model better exploit its training data, we train EDITOR on $x\% \in [0, 100]$ of the data, and compare using random samples with samples ranked by increasing edit distance.Figure 5.6 shows the number of updates to convergence on the development dataset for controllable simplification with/without CL. Training converges early (70 iterations only) on 13% of the easiest samples with oracle edit distance between the input and the output sequence $<= 2$. This supports the hypothesis that despite adding noise, our approach yields easier examples to train on. The order in which samples are presented matters, as adding batches with larger edit distance ($> 63\%$ data) without maintaining the order of the samples converges early. By contrast, the curriculum pacing function adds samples in order of increasing difficulty, allowing the model sufficient training time to learn from new samples while improving overall performance across metrics. Training with curriculum reduces the overall loss consistently, leading to better generalization (Figure 5.7 on the right).

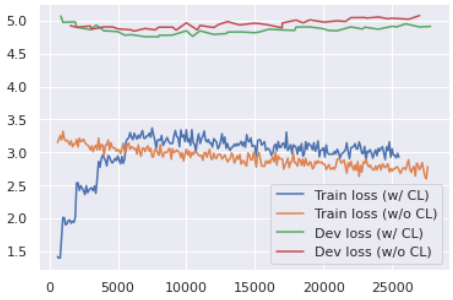
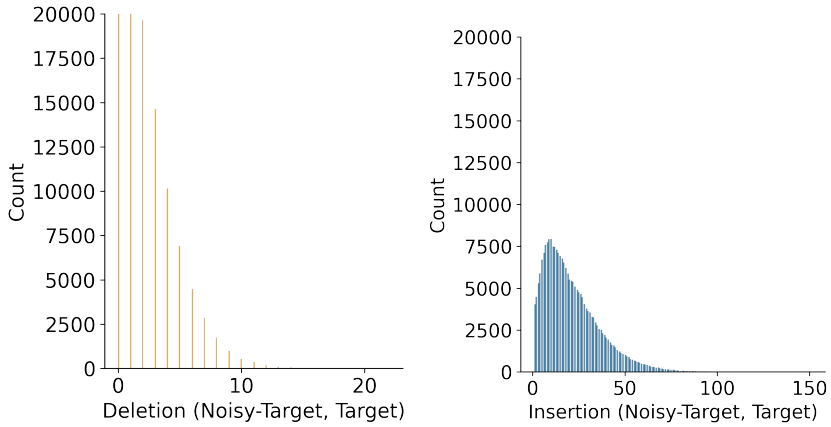
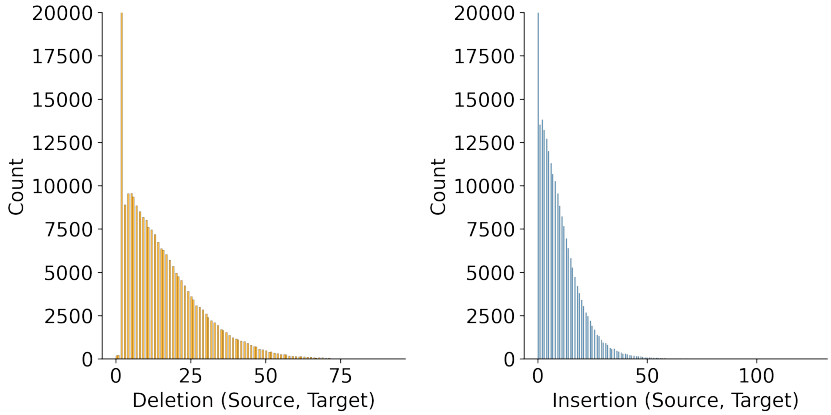


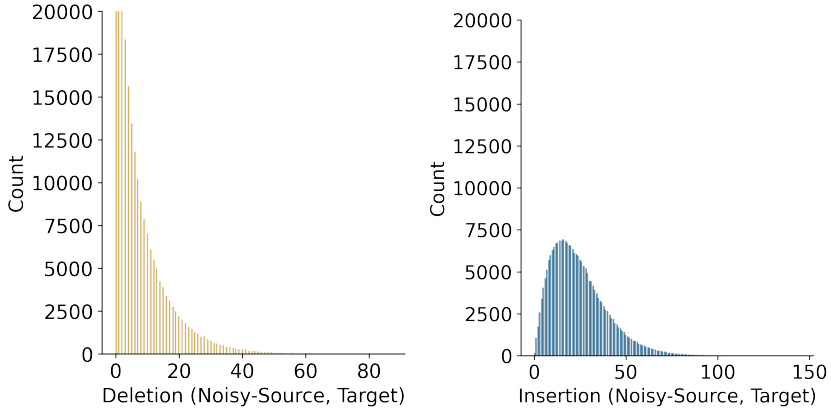
Figure 5.7: Training with curriculum reduces the loss on the development dataset leading to better generalization on Controllable TS.



(a) EDITOR ś roll-in (Training)



(b) Inference Distribution



(c) EDITING roll-in (Training)

Figure 5.5: Distribution of Oracle Edit Operations (*Insertions*/*Deletions*) observed on Controllable TS. Our proposed roll-in policyś distribution of edit operations is closer to the inference distribution, while enabling exploration during training.

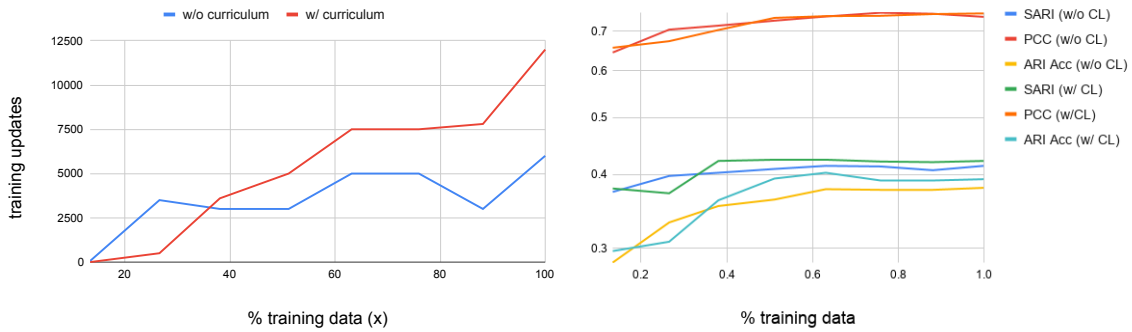


Figure 5.6: The sample order during training matters as training without curriculum on the same amount of data ($>= 40\%$) converges early (plot on the left) and to lower performance across all metrics (plot on the right) relative to training with curriculum using the same data.

CRITERIA	SARI	PCC	% ARI-ACC	CORR.
Random	40.7	0.749	38.6	-
Length Ratio	41.0	0.762	39.0	0.26
Grade Difference	40.7	0.730	38.3	0.19
EDITCL	42.0	0.758	39.6	1.00
- EDITING roll-in	40.1	0.734	37.8	-
- CL	41.2	0.742	39.3	-

Table 5.8: On Newsela-grade dev dataset: Using Edit distance as the difficulty criteria improves over both task-specific (Grade Difference) and task-agnostic (Length ratio) criteria. Our proposed EDITING roll-in and curriculum-controlled roll-out provides complementary advantages to the model training.

Ranking Criteria We compare the edit-distance (EDITCL) with other curriculum criteria in Table 5.8 where the order of examples is a) random, b) controlled by the length ratio between source and target sequence (Length Ratio), c) governed by the difference between the source and target grade levels (Grade Difference). Our proposed criterion outperforms both task-specific (`Grade Difference`) and task-agnostic criteria (`Length Ratio`) on the Newsela Grade development set across all the metrics. `Length Ratio` achieves better correlation with Edit distance than `Grade Difference` which is also reflected by its performance (SARI: +0.3, PCC: 0.032, ARI: 0.7) on the Controllable Simplification task. This might reflect the fact that higher grade differences do not necessarily require more

edits to be performed, for instance when the sentence to be simplified is already relatively simple. These mismatches do not occur when the edit distance itself is used as the sample difficulty criterion.

Complementarity of roll-in and roll-out design We report the performance of the `From Input` model, when trained with curriculum only without the EDITING policy, i.e. `EDITCL- EDITING` in the same Table 5.8. Both `EDITING` roll-in and curriculum controlled roll-out provides complementary advantages to the model training as removing either results in the drop in performance across all the metrics for controllable TS. However, we observe larger drop in the scores when we do not apply the EDITING policy which shows that our proposed roll-in policy is necessary to reap the benefits of curriculum learning.

5.6 Conclusion

We contrast two approaches that inject guidance during inference or training when adapting NAR models to Controllable Text Simplification. The inference strategy relies on lexical complexity signals derived from data statistics. For training, we propose two complementary strategies to make the best use of the diverse training examples during training: 1) a new roll-in policy that generates intermediate sequences that the model is likely to encounter during inference and 2) a curriculum to control the difficulty of the roll-out policy throughout training. Together, these strategies improve the quality of the outputs and better match the desired target grade levels than injecting lexical complexity signals during inference.

Chapter 6: Controllable Text Simplification Using Pre-trained Models

In the previous chapter, we present edit-based models that exhibit finer-grained control over the generated simplified texts by learning to directly model text editing operations (like insertions and deletions) for specific target grade levels. An alternative approach proposed in concurrent work leverages large-scale unsupervised pre-training and low-level side constraints (Sheang and Saggion, 2021; Martin et al., 2022) for controlling the properties of the generated outputs. These side constraints are appended to the input as special tokens that encode desired properties of the output and the pre-trained models are then fine-tuned on supervised samples consisting of input text augmented with control tokens, paired with output text. The low-level control tokens are designed to directly specify the nature of the text transformation operation to be performed such as length, degree of paraphrasing, lexical complexity, and syntactic complexity. Figure 6.1 shows one example where the low-level control tokens are the ratio of the number of words (W) between the source and the target and the maximum dependency tree depth (DTD) between the source and the target text. For an original complex text at grade 8, when simplifying to grade 6, the low-level control values indicate a conservative rewrite, whereas, for grade 3, the properties encoded by the control tokens reflect a relatively more lexical and structural change.

While custom edit-based architectures directly learn to map target grade levels and the source text to low-level edit operations like insertions/deletions (Chapter 5), it is unclear how the low-level control values should be defined and set in practice in the latter setting for grade-specific text simplification. And, these low-level control tokens might be too

Original: Paracho, the "guitar capital of Mexico," makes nearly 1 million classical guitars a year, many exported to the United States.

Grade 5: Paracho **is known as** the "guitar capital of Mexico." The town makes nearly 1 million classical guitars a year, with many exported to the United States.

`Word Length Ratio (W):` 1.19

`Dependency Tree Depth Ratio (DTD):` 1.50

Grade 3: Paracho **is known as** the "guitar capital of Mexico." The town makes **many guitars and sells some** in the United States.

`Word Length Ratio (W):` 0.96

`Dependency Tree Depth Ratio (DTD):` 1.00

Figure 6.1: Simplified texts can be obtained by either specifying the target audience (via grade level) or by using low-level control tokens to define the TS operation to be performed relative to the complex text (W, DTD).

cumbersome and counterintuitive to be set by teachers and readers. Prior work sets and uses these low-level control values at the corpus level during inference by searching for control token values on a development set. This is done via maximizing a utility computed using an automatic evaluation metric, SARI, a metric designed to measure lexical simplicity (Xu et al., 2016). While appealing in its simplicity, the model is never evaluated for *control* at the instance level as the control token values are always set at the corpus level.

In this chapter, we present a systematic empirical study of the impact of control tokens on the degree and quality of simplifications achieved at the instance level as measured by automatic text simplification metrics. Our study shows that most corpus-level control tokens have an opposite impact on adequacy and simplicity when measured by BLEU and SARI respectively. As a result, selecting their values based on SARI alone yields simpler text at the cost of misrepresenting the original source content. To address this problem, we introduce simple data-driven solutions to predict what control tokens are needed for a given input text and a desired grade level, based on surface-form features extracted from the source text and the desired complexity level. Our findings demonstrate that using predicted

low-level control tokens leads to a wider range of edit operations and improves the simplicity of the generated outputs.

6.1 Controllable Text Simplification: Background

While text simplification has been primarily framed as a task that rewrites complex text in simpler language in the NLP literature (Chandrasekar et al., 1996b; Coster and Kauchak, 2011b; Shardlow, 2014; Saggion, 2017; Zhang and Lapata, 2017), in practical applications, it is not sufficient to know that the output is simpler. Instead, it is necessary to target the complexity of the output language to a specific audience. Acknowledging that text simplification is highly audience-centric and source-text dependent (Stajner, 2021), recent work has focused on developing techniques to *control* the degree of simplicity of output using various levels of varying granularity. Usually, the control mechanism in a generative model is induced using side constraints (Sennrich et al., 2016a) that are special tokens that can be appended to the source or the target sequence to control the properties of generated outputs.[1]

Control Token Mechanisms We provide an overview of the type of the control tokens introduced in prior work for text simplification in Tables 6.1 and 6.2. Coarse-grained control over the degree and the nature of the simplification, e.g. via source and target grade levels can be considered more interpretable for end users (Table 6.1, `[1-6, 12]`), whereas controlling multiple low-level attributes (Table 6.1, `[7-11]`) that map text simplification operations to specific properties of the input and the output text can provide much better control over the generated text.

[1] It is important to note that the term "control" used in this work does not imply strict constraint enforcement on the model's output. Rather, the control tokens or side constraints merely serve as inputs that influence the model's behavior and encourage the generation of outputs with desired properties.

PAPER-ID	PAPER	CONTROL TOKENS	HOW TO SET?
[1]	Scarton and Specia (2018a)	Target Grade and/or Operations	User-defined or Predicted
[2]	Nishihara et al. (2019)	Target Grade (TG)	User-defined
[3]	Agrawal et al. (2021b)		
[4]	Yanamoto et al. (2022)		
[5]	Zetsu et al. (2022)		
[6]	Agrawal and Carpuat (2022)	+ Source Grade	
[7]	Martin et al. (2019)	ACCESS {C, L, WR, DTD}	Corpus-level Optimization with SARI
[8]	Sheang and Saggion (2021)	+ W	
[9]	Martin et al. (2022)	− L + RL	
[10]	Maddela et al. (2021)	CC	Average over the training dataset
[11]	Qiao et al. (2022)	10 Psycho-linguistic Features	Corpus-level Optimization with SARI
[12]	Kew and Ebling (2022)	λ per grade-level	Hyperparameter Tuning with SARI

Table 6.1: How are control tokens defined and set in prior work?

ID	NAME	DESCRIPTION
C	NbChars	character length ratio between source and target.
L	LevSim	character-level Levenshtein similarity (Levenshtein et al., 1966) between source and target.
WR	WordRank	ratio of log-ranks (inverse frequency order) between source and target.
DTD	DepTreeDepth	maximum depth of the dependency tree of the source divided by that of the target.
W	NbWords	word length ratio between source and target.
RL	Replace-only LevSim	character-level Levenshtein similarity only considering replace operations between source and target.
CC	Copy Control	percentage of copying between source and the target

Table 6.2: Description of Control Tokens

However, it is often unclear how those low-level control values should be defined during inference as these could vary significantly based on the **source text** and **the degree of the simplification** required. In all prior work (Martin et al., 2020a, 2022; Sheang et al., 2022; Qiao et al., 2022), these values are set and evaluated at the corpus level. This is achieved by doing a hyperparameter search, optimizing for a single metric SARI on the entire validation set. SARI measures the lexical simplicity based on the n-grams kept, added, and deleted by the system relative to the source and the target sequence. We identify two key issues with this corpus-level search-based strategy for setting control values as described below:

Input Agnostic Control Setting these control values at the corpus level disregards the nature and complexity of the original source text. It does not account for what can and should be simplified in a given input (Garbacea et al., 2021) and to what extent. We show that the control values are indeed dependent on all these factors as exhibited by a large variation observed in the values of the control tokens at the individual target grade levels (Figure 6.7).

Costly Hyperparameter Search Searching for control tokens value at the corpus-level is an expensive process. Martin et al. (2022) use the OnePlusOne optimizer with a budget of 64 evaluations using the NEVERGRAD library to set the 4 ACCESS hyperparameters (upto 2 hours on a single GPU). Sheang et al. (2022) select the values that achieve the best SARI on the validation set with 500 runs. This takes >= 3 days when training the model takes only 10-15 hours. As these values are domain and corpus-specific, optimizing these values even at the corpus level for multiple datasets is computationally expensive.

We provide an analysis of the impact of these control values defined at the corpus level on the degree and nature of TS performed at the instance level in the next section.

6.2 How do Control Tokens Impact TS?

Study Settings We study the impact of setting the low-level control values at the *corpus level* on the *instance-level* simplification observed using automatic text simplification metrics. We conduct our analysis on the Newsela-grade dataset (§ 4.3.2). We use the control tokens defined in Sheang et al. (2022) (see Table 6.2 `[1-5]`) added to the source text as a side constraint in the format, `W_{} C_{} L_{} WR_{} DTD_{} {Source_text}`.

Instance-Level Findings Following prior work (Martin et al., 2019), we select the control tokens that maximize SARI on the Newsela-grade development set. We measure the

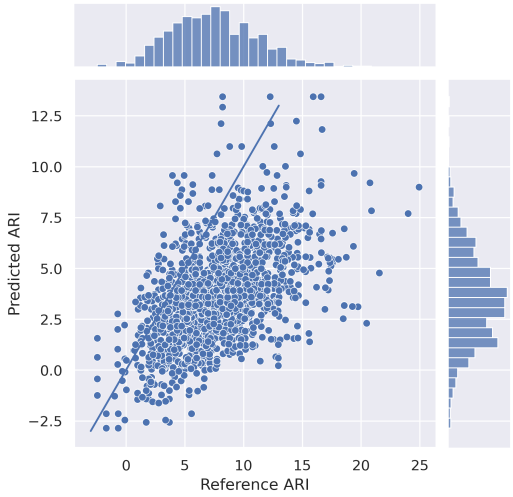


Figure 6.2: ARI accuracy on the Newsela-grade Development Set: 12%. Setting corpus-level control values results in over or under-simplification.

complexity of the outputs generated and compare them with the complexity of the Newsela references by computing the Automatic Readability Index (Equation 4.4).

As can be seen in Figure 6.2, control tokens set at the corpus level using SARI tend to over or under-simplify individual input instances. When the reference distribution exhibits diversity in the degree of the simplification performed, setting corpus-level values is sub-optimal: outputs are frequently over- or under-simplified as illustrated by the difference in the reference and predicted grade levels.

Corpus-Level Findings Figure 6.3 shows the correlation between 100 sets of control values set at the corpus level and automatic TS metrics computed using the outputs generated: SARI, BLEU, and FR (Flesch Reading Ease): Most control tokens have an opposite impact on SARI and BLEU, except the character length ratio, `(C)`. This suggests that setting their values by optimizing for SARI alone at the corpus level can be misleading, as a high SARI score can be achieved at the expense of a lower adequacy score. These findings are consistent with the observation of Schwarzer and Kauchak (2018) who note a similar negative correlation between human judgments of simplicity and adequacy and caution that:

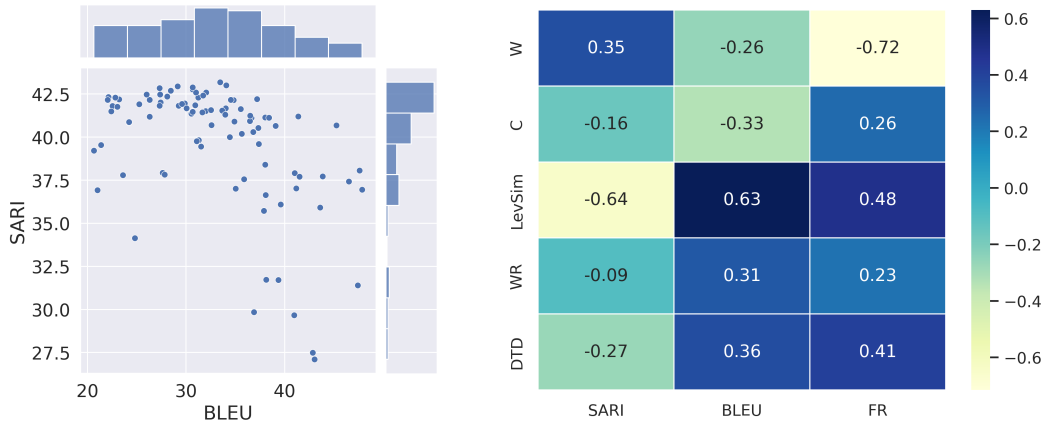


Figure 6.3: Adequacy-Simplicity Tradeoff on the Newsela-Grade development set when using different control tokens set at the corpus-level.

"improvement in one metric and not the other may be due to this inverse relationship rather than actual system performance". These results lead us to concur with the recommendation of Alva-Manchego et al. (2021), which advocates for always augmenting SARI with an adequacy metric for text simplification evaluation.

Overall, this analysis highlights important limitations of the simplification abilities provided by setting control tokens based on optimizing SARI at the corpus level. We propose instead a simple method to predict these values based on each input instance and the desired output complexity.

6.3 Grade-Specific Text Simplification with Instance-Level Control

Since the simplification for a given instance should depend on the original source text, its complexity, and the desired target complexity, we introduce a Control Predictor module (CP) that predicts a vector of control token values V for each input X at inference time. Figure 6.4 shows the overall inference pipeline for generating the simplified text using the control token values predicted using CP.

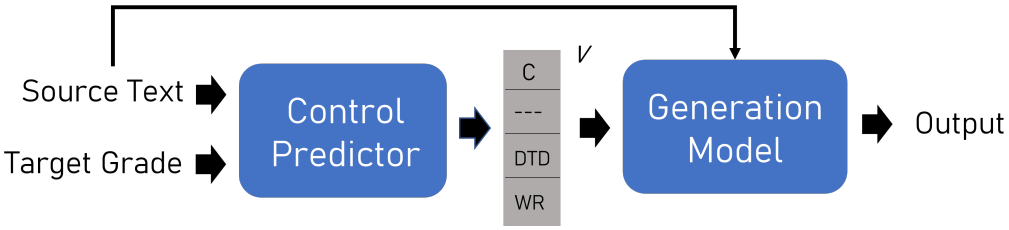


Figure 6.4: Inference Pipeline for Generating Simplified Text with Control Predictor.

Predicting Control Tokens We train the Control Predictor (CP(θ): $X \rightarrow V$) to predict the control vector given features extracted from an input text and the input and output grade levels. Let $\{x_i, y_i\} \in D$ represent a complex-simple pair, and the ACCESS controls associated with this pair be V_i. For example, $V_i = \{W_i, C_i, L_i, WR_i, DTD_i\}$ in Sheang et al. (2022). We propose both single and multi-output regression solutions for predicting V as described below:

a) **CP-Single:** The model is trained to predict the individual control tokens, resulting in one model per control value.

b) **CP-Multi:** The model is trained to optimize the mean RMSE error over all the control dimensions.

We train a simple feature-based Gradient Boosting Decision Trees classifier to predict the control values, V using the CatBoost library,[2] using several surface-form features extracted from the source text as described below:

a) Number of Words

b) Number of Characters

c) Maximum Dependency Tree Depth

d) Word Rank

e) Mean Age of Acquisition (Schumacher et al., 2016)

[2] https://catboost.ai/en/docs/

TS Model Training Given a source text (x) and a control vector v, the controllable TS model $P(y|x, v)$, is trained to generate the reference simplified output that conforms to v in a supervised fashion, by setting v to oracle values derived from the reference and optimizing the cross-entropy loss on the training data.

6.4 Experimental Settings

6.4.1 Data and Automatic Evaluation Metrics

Data We use the Newsela-grade dataset (Section 4.3.2) that includes 470k/2k/19k samples for training, development and test sets respectively.

Metrics We automatically evaluate the truecased detokenized system outputs using:

a) **SARI** (Xu et al., 2016), which measures the lexical simplicity based on the n-grams kept, added, and deleted by the system relative to the source and the target. [3]

b) **BERTSCORE** (Zhang et al., 2019) for assessing the output quality and meaning preservation

c) **ARI-Accuracy** (Heilman et al., 2008) that represents the percentage of sentences where the system outputs' ARI grade level is within 1 grade of the reference text.

d) **%Unchanged Outputs (U)** The percentage of outputs that are unchanged from the source (i.e., exact copies).

We evaluate the fit of the control predictor in predicting V using **RMSE** and **Pearson Correlation** with the gold ACCESS values.

[3] https://github.com/feralvam/easse

6.4.2 Model Configurations

We finetune the T5-base model following Sheang et al. (2022) with default parameters from the Transformers library except for a batch size of 6, maximum length of 256, learning rate of 3e-4, weight decay of 0.1, Adam epsilon of 1e-8, 5 warm-up steps, and 5 epochs. For generation, we use a beam size of 8. The learning rate and the tree depth are set at 0.1 and 6 respectively for training all the control predictor models.

We use a learning rate of 0.1 and a tree depth of 6 for training all the control predictor models which takes approximately 5-10 minutes.

Controllable TS Variants We compare the prediction-based TS models above with two variants:

- GRADE TOKENS: a model that uses high-level control token values, i.e. the source grade (SG) and the target grade (TG) levels when finetuning the generation model (Scarton and Specia, 2018a).
- AVG-GRADE: a simple approach that sets control values with the average of the values observed for the source-target grade pair.

Controllable TS Baselines We compare our approach with the corpus-level hyperparameter search strategy (CORPUS-LEVEL) used in prior work that selects the best low-level control values based on SARI only (Martin et al., 2020a).

Source Grade at Inference While the desired target grade level is known during inference, we automatically predict the grade level of each source sentence using the ARI score in all the settings.

6.5 Results

We first discuss the accuracy of the Control predictor in estimating the ACCESS control values on the Newsela-Grade dataset and then show the impact of using the predicted control tokens as constraints towards controlling the degree of simplification in the generated outputs.

6.5.1 Intrinsic Evaluation of Control Predictor

CONTROL	CORRELATION(↑)					RMSE (↓)
	W	C	LevSim	WR	DTD	
CP-SINGLE	0.405	0.407	0.567	**0.398**	0.567	0.197
CP-MULTI	**0.420**	**0.422**	**0.568**	0.393	**0.570**	0.196

Table 6.3: `CP-Multi` improves correlation with ground truth ACCESS control token values (`W, C`) on Newsela-grade over `CP-Single`.

Table 6.3 shows the correlation and RMSE of predicted values with gold low-level control tokens. Training the model to jointly predict all values, V improves correlation ($+0.015$) for `W, C` over training independent models (`CP-Single`) for individual control tokens. We show that this can be attributed to the correlation amongst the target control values in Figure 6.5. Both `(W, C)` exhibit moderate correlation with `DTD`. There is a drop in correlation for `WR` when training a joint model which is expected as `WR` is a proxy for lexical complexity and is the most independent control token.

The correlation scores for the control tokens `(W, C, DTD, LevSim, WR)` range from -0.30 `(S_W, W)` to 0.40 `(TG, LevSim)` after removing unchanged examples (simplified text is same as the original text). The moderate correlation between the source features (prepended with S) and the low-level tokens suggests that the source text influences the nature of simplification that can be performed. Additionally, `LevSim` controls the

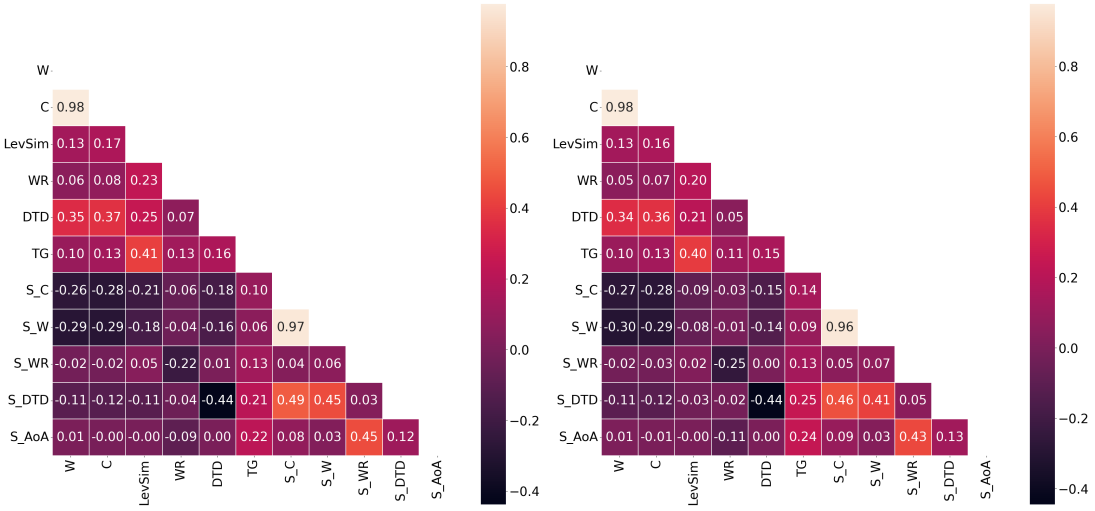


Figure 6.5: Correlation between source features and control token values on the Newsela-Grade training set before (left) and after (right) removing unchanged examples.

degree of simplification as suggested by its moderate correlation with the target grade level and can be considered the most prominent token for balancing the adequacy-simplicity tradeoff.

6.5.2 Overall Grade-Specific TS Results

We show how the different control tokens as side constraints influence the degree and the nature of simplification in the generated outputs in Table 6.4.

Setting corpus-level control for grade-specific TS is suboptimal. Optimizing SARI alone for selecting the low-level control tokens and setting corpus-level control values is suboptimal for matching the desired complexity level. This is indicated by the low ARI accuracy of only 3.1%.

Predicted instance-level control outperforms grade or corpus-level control. Predictor-based models (`CP-Single, CP-Multi`) that set control tokens for each instance based on source features improve simplicity scores compared to using `Avg-Grade`, which only

CONTROL	BERTSCORE	SARI	%ACC	%U
Low-level				
CORPUS-LEVEL	0.922	42.19	3.1	0
AVG-GRADE	0.945	44.09	32.7	0
CP-SINGLE	0.946	45.54	36.3	2
CP-MULTI	0.946	**45.65**	36.3	3
High-level				
GRADE TOKENS	**0.955**	43.02	**39.8**	34
ORACLE	0.955	53.53	56.7	12

Table 6.4: Results on the Newsela-grade dataset: using source-informed tokens (CP-$*$) significantly improves SARI over alternative control mechanisms. All differences are significant except the difference between CP-Single and CP-Multi with a p-value of 0.00.

uses grade information to set control values. These models show improvements in SARI (+1.4-1.5) and ARI (+3.6%) scores, highlighting the importance of setting control tokens at the instance level rather than relying solely on just the grade information. Furthermore, setting control tokens based on the average values observed for a given source-target grade pair, i.e., `Avg-Grade` significantly improves both BERTSCORE and ARI-based metrics across the board compared to the `Corpus-level` approach.

Grade-level (high) and operation-specific (low) control tokens exhibit different adequacy and simplicity tradeoffs. Low-level control tokens offer more precise control over the simplicity of outputs, resulting in improved SARI scores by at least 2 points compared to `Grade Tokens`. However, this advantage comes at the cost of lower adequacy (BERTScore) and control over desired complexity (ARI Accuracy). The models trained with low-level control values exhibit lower grade accuracy scores partly due to the limited representation of the need for text simplification (Garbacea et al., 2021) during the generation process as suggested by a lower percentage of exact copies in the output compared to

`Grade Tokens` (Exact copies in references: 12%). On the subset of the test set with no exact matches between the source and the reference text, `Grade Tokens` and `CP-Multi` receive ARI accuracy of 34.2 and 34.0 respectively. Furthermore, we hypothesize that the models trained with low-level control exhibit low meaning preservation because none of the control tokens directly encourage content addition during text simplification. And, while the model learns to perform an appropriate content deletion, it does not generate a fitting substitution or addition as required to preserve the meaning of the original source text. We show a detailed operation-specific analysis in the following section.

6.5.3 Impact of Control Tokens on TS Edit Operations

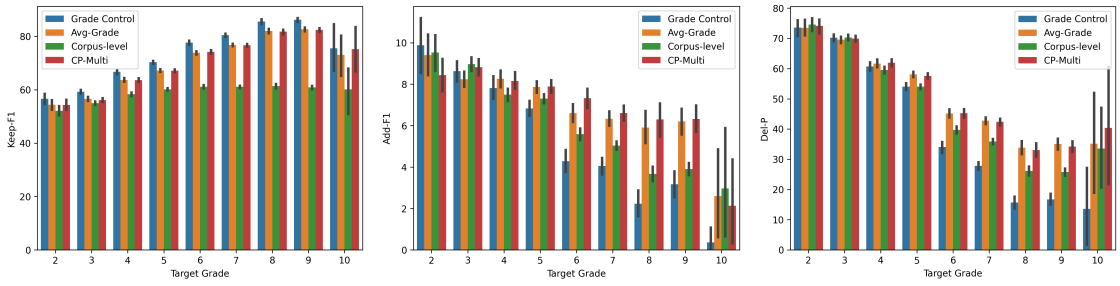


Figure 6.6: Edit Operations by Target Grade Levels: CP-Single performs correct and diverse edits as suggested by the high Add-F1 and Del-P scores for all target grade levels > 4.

Predicting control tokens for individual instances improves coverage over the range of control values exhibited by the oracle. We show the distribution of control values observed by different ways of setting the low-level control values across all target grade levels in Figure 6.7. The high variance in the range of oracle values confirms that the control tokens vary based on the source texts' complexity and desired output complexity. Where `Corpus-level` fails to even match the mean distribution, `CP-Multi` is able to cover a wider range of control values.

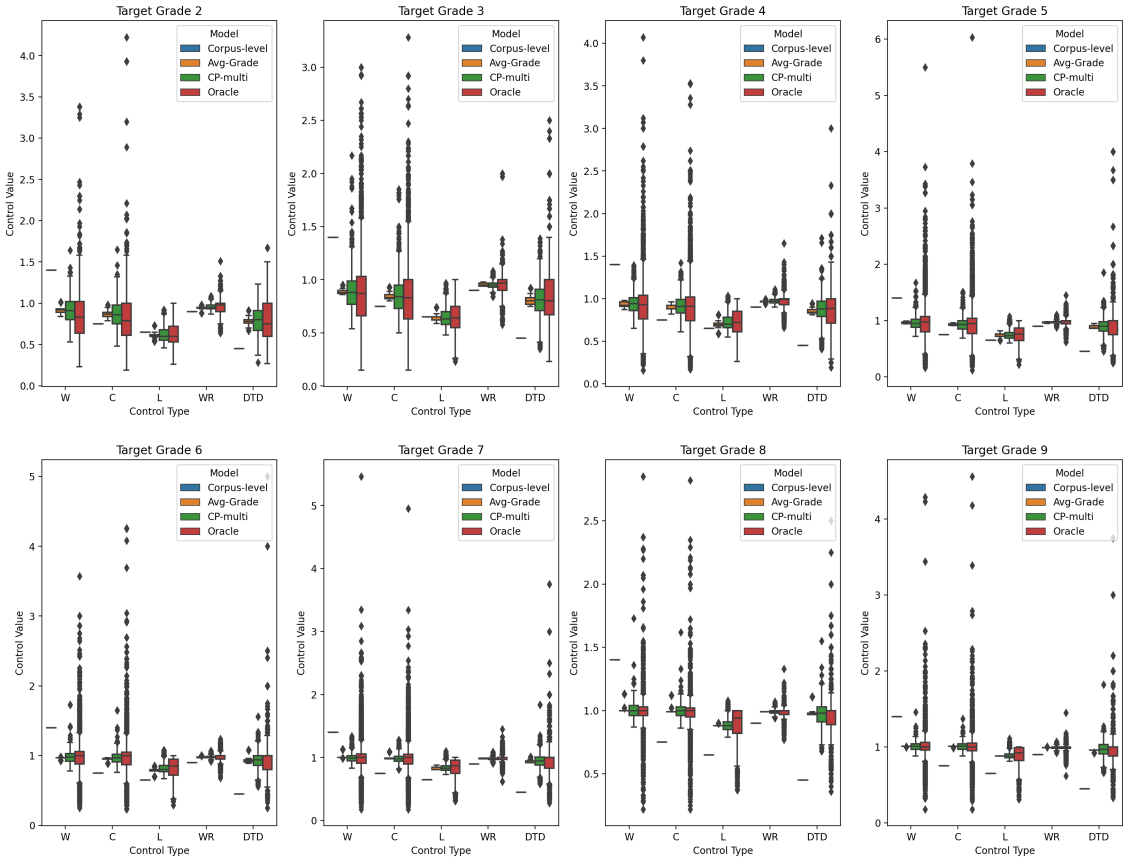


Figure 6.7: Distribution of control token values for different model variants by Target Grade level.

Simplified outputs generated using predicted control tokens exhibit diverse edit operations. Figure 6.6 shows the distribution of the KEEP-F1, DEL-P, and ADD-F1 scores by target grade level for the models trained with different control types, where ADD-F1 computes the F1 score for the n-grams that are added to the system output relative to the source and the reference text. The model's deletion capability is measured by the F1 score for n-grams that are kept (KEEP-F1) and the precision of deletion operation (DEL-P) with respect to the source and the reference.

`CP-Multi` consistently achieves better or competitive DEL-P across all target grade levels over alternative control mechanisms, suggesting that setting control values informed by both the source and desired complexity level improves the model's ability to appropriately

delete redundant information. The former also generally improves ADD-F1 scores, highlighting that the model also appropriately performs lexical substitution or content addition as required across different grade levels (except grades 2 and 10). Additionally, low-level control tokens (`CP-Multi`, `Avg-Grade`) produce more correct diverse modifications than high-level control (`Grade Tokens`) as implied by the better ADD-F1 and DEL-P scores across grade levels > 3, where the latter is more conservative (high KEEP-F1), hence generating more meaning-preserving simplifications.

6.6 Conclusion

In this chapter, we present a systematic analysis of the impact of control tokens set at the corpus level on the degree and quality of simplification achieved by controllable text simplification models at the instance level. Our findings show that control tokens exhibit an opposite correlation with adequacy and simplicity. Hence, selecting their values at the corpus level based on SARI alone leads to over or under-simplifying individual instances. This motivates a new approach to set low-level control tokens during inference by predicting them given a source text and desired target grade level. We show that this approach is effective at improving the quality and controlling the degree of simplification in generated outputs based on automatic evaluation. Furthermore, the predicted low-level control tokens yield more diverse edit operations than alternative ways of setting control on the Newsela-grade dataset. However our evaluation so far was limited to automatic metrics, in the next Chapter, we turn to human evaluation of the controllable TS systems.

Chapter 7: A Framework for Evaluating Correctness of Simplified Texts

Simplified texts that are easier to read are not helpful if they mislead readers by providing incorrect information. Assessing to what extent the meaning of the original text is preserved should therefore be a critical dimension of TS evaluation (Stajner, 2021), and a pre-requisite to determining whether and how TS can be used in practice. Additionally, evaluating individual sentences out of context may not be sufficient to establish the correctness of model-generated texts, as human simplifications often occur at the document or the passage level (Devaraj et al., 2022). Yet, the correctness of TS outputs is primarily evaluated intrinsically, with automatic metrics that compare system outputs with human-written reference simplifications and/or the original source (Papineni et al., 2002a; Xu et al., 2016; Maddela et al., 2022), or with generic human assessments of simplicity and accuracy of individual sentences performed outside of a context of use (Schwarzer and Kauchak, 2018). While these evaluations can guide model development, they do not directly address the question of whether readers get incorrect information from the text.

In this work, we conduct a human evaluation of the correctness of state-of-the-art TS systems by measuring the extent to which people can answer questions about key facts from the original text after reading a simplified version. We design reading comprehension tasks (RC) to directly assess TS correctness, different from prior uses of reading comprehension to assess people's reading efficiency (Angrosh et al., 2014; Laban et al., 2021). This framework lets us conduct a controlled comparison of the correctness of simplified texts, whether written by humans or by TS systems: we compare people's ability to answer questions

about the original text, a human-written simplified version, and nine TS-generated versions which represent a diverse set of supervised and unsupervised approaches from the recent TS literature. Furthermore, we investigate how existing automatic TS evaluation metrics and automatic question answering systems approximate the human judgments of correctness we obtained.

7.1 A Reading Comprehension-based Human Evaluation Framework

Background Reading comprehension tests are standard tools used by educators to assess readers' understanding of text materials, and thus provide an assessment of TS that is more in line with its intended use. They have been used to show that human-simplified texts are easier to comprehend by L2 learners (Long and Ross, 1993; Tweissi, 1998; OH, 2001; Crossley et al., 2014; Rets and Rogaten, 2021), as well as secondary and post-secondary students (Heydari et al., 2013). For instance, Long and Ross (1993) conduct a reading comprehension study with 483 Japanese students with varying English language proficiency levels and found that participants who had access to linguistically simplified or elaborated texts scored higher on the RC tasks than those who read the original text. Similarly, Crossley et al. (2014) showed a linear effect of text complexity on comprehension even when accounting for individual language and reading proficiency differences as well as their background knowledge. Rets and Rogaten (2021)'s study with 37 adult English L2 users showed better comprehension and faster recall for participants with low English proficiency levels with simplified texts. Finally, in a within-subject study with four original and four simplified texts involving 103 participants with varying levels of English proficiency (beginner to native), Temnikova and Maneva (2013) show that utilizing Controlled Language (Temnikova et al., 2012) for TS improves reading comprehension. All these studies are conducted at

the paragraph or the document level based on human-written simplifications which are implicitly assumed to be correct.

The use of reading comprehension for evaluation of automatically simplified texts is more limited and has focused on measuring how efficiently people read TS outputs. Angrosh et al. (2014) first used reading comprehension to evaluate *automatically* simplified texts from multiple TS models, with non-native readers of English. They conduct a multiple-choice reading comprehension test using five news summaries chosen from the Breaking News English website, originally at reading level 6 (hard) and simplified manually or via automatic TS systems. Their study found no significant differences between the comprehension accuracy of different user groups when reading automatically generated simplifications. However, they note that the drop in comprehension scores for some of these systems could be accounted to the content removal which can make some questions unanswerable. Hence, it is not clear whether the differences are non-significant due to user understanding, errors introduced by TS systems, or the effectiveness of simplifications. Laban et al. (2021) also conduct a reading comprehension study to evaluate the usefulness of automatic TS outputs with automatically generated reading comprehension questions that can be answered by original text and human-written references. They found that shorter passages generated by automatic TS systems lead to a speed-up regardless of simplicity. However, the automatic generation of questions mostly limited them to factoid, thus limiting the scope of understanding tests, and it is unclear whether the TS errors could render the RC questions unanswerable.

In this work, we design a reading comprehension task to assess the correctness of TS via carefully constructed multiple-choice questions, and use it to conduct a thorough controlled evaluation of a diverse set of state-of-the-art TS systems.

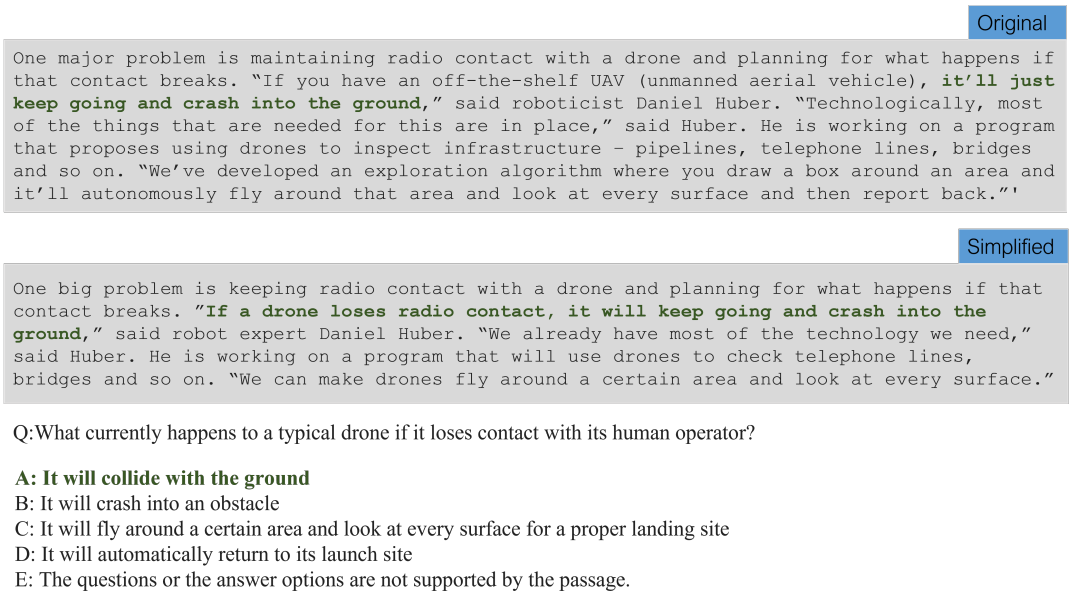


Figure 7.1: A Reading Comprehension question to be answered after reading either the original or the simplified text. The answer options include the correct answer (A), three incorrect options of varying difficulty (B,C,D), and an option (E) that capture questions rendered unanswerable after automatic TS.

Overview Our human evaluation is based on the following task: participants are presented with text and then are asked questions to test their understanding of some of the information conveyed in the text, as illustrated in Figure 7.1. We seek to measure whether participants who read simplified versions of the original paragraph can answer questions as well as those who read the original. However, our goal is not to assess the participants but TS systems: when working with participants who are proficient in the language tested, we assume that differences in reading comprehension accuracy indicate differences between the quality of TS systems that produce the different simplifications.

OneStopQA Within this simple framework, the design of the RC questions and answers is critical to directly evaluate the correctness of automatic TS systems. We build on the OneStopQA reading comprehension exercises (Berzak et al., 2020), which is well suited to our task since it targets the real-world need of supporting readers with low English

proficiency, and there is already evidence that it is a sound instrument to capture differences in reading comprehension from human-written text.

Specifically, OneStopQA is based on texts from the onestopenglish.com English language learning portal (Vajjala and Lučić, 2018), which are drawn from The Guardian newspaper. Questions and answers are curated using STARC, an annotation framework designed for reading comprehension that combines structured answers with span annotations for both correct answers and distractors (Berzak et al., 2020). The structured answers reflect four fundamental types of responses, ordered by miscomprehension severity: option A indicates correct comprehension, option B shows the ability to identify essential information but not fully comprehend it, option C reflects some attention to the passage's content, and option D shows no evidence of text comprehension. This structure helps support analyses by examining specific types of miscomprehension. Participants are presented with the answer options in a randomized order to minimize any potential bias or pattern recognition that they may use to guess answers. The correct answer is not present verbatim in the critical span, a text span in the passage upon which the question is formulated. We note that the questions only target a subset of the information conveyed in a passage, and hence, our evaluation framework does not provide a measure of completeness.

Further, prior work suggests that OneStopEnglish text and OneStopQA questions provide a sound basis for evaluating automatic TS, as they can capture differences in reading comprehension from manually simplified text: Gooding et al. (2021) found a statistically significant difference between users scrolling interactions and the text difficulty level in a 518-participant study and Vajjala and Lucic (2019) showed that the nature of the reading comprehension questions can impact text understanding.

Targeting Answerability We augment the OneStopQA answer candidates with a fifth option motivated by the failure modes of automatic TS. For each question, participants have

the option to pick "unanswerable" (UA), which they are instructed to select when "The questions or the answer options are not supported by the passage.". This lets us directly measure how often readers judge that there is no support for answering the question based on the input text, which is a more salient problem when presenting participants with automatic than human-written simplifications. The resulting reading comprehension problems are illustrated in Figure 7.1.

Text Granularity Participants are presented with a paragraph of text before answering each question, thus moving away from the prior focus on evaluating TS at the sentence level. In real world settings, people are unlikely to use text simplification on isolated sentences and might be able to understand important information by making inferences from the context. Thus evaluating text simplification outputs at the paragraph level strikes a good balance between providing a realistic amount of context to readers without making the task too long.

Measures Given M paragraphs from one of the following: original text, human-written simplification, or the nine automatic TS systems, each paragraph $P \in M$, is accompanied by a set of Q questions with the 5 multiple-choice answers $\{q, a_1^5\}_1^q$, as described above. We measure the **correctness** (C) of the simplified texts using the number of questions answered correctly for that system by human participants. Formally,

$$C_S = \frac{1}{M \times Q} \sum_{m=1}^{M} \sum_{q=1}^{Q} 1[Selected == Correct] \tag{7.1}$$

where Selected is the answer marked by human participants for a given passage P. We rank the automatic TS systems based on the ranking induced by C. Systems with higher C scores produce simplifications that help people answer reading comprehension questions correctly.

We compute **answerability** (Ans) using questions that were marked "UA" by the participants as:

$$Ans = 1 - \frac{1}{M \times Q} \sum_{m=1}^{M} \sum_{q=1}^{Q} 1[Selected == UA] \tag{7.2}$$

Systems often produce outputs that do not support answering the question, perhaps due to over-deletion or other serious output pathology (Devaraj et al., 2022). Systems with high Ans scores produce outputs that are not necessarily correct but still support answering the questions.

7.2 Experimental Setup

First, we describe the experiment details including data, participant selection and study design. Then, we outline the TS systems that we selected for evaluation.

7.2.1 Study Design

Data The OneStopQA dataset includes 30 articles containing 162 paragraphs in total at three difficulty levels: Elementary, Intermediate, and Advanced. Each passage is accompanied by three multiple-choice questions that can be answered at all levels of difficulty. The simplified versions include common text simplification operations such as text removal, sentence splitting, and text rewriting. We select the first two paragraphs from each of the 30 articles and associated questions resulting in 60 unique passages and 180 questions in total.

Participants The participants are paid directly through the crowd-sourcing platform at an average rate of USD 15/hour. The task is conducted on the Prolific crowd-sourcing platform.[1], we recruit 112 native speakers of English between ages 18-60 identified by their

[1] prolific.co

first language and with an approval rating of at least 80% for evaluating the correctness of TS systems.

Task Design Each participant is provided with the following instruction: *In this study, you will be presented with 6 short excerpts of English text, accompanied by three multiple-choice questions. You are asked to answer the questions based on the information presented in the text.*. A participant is presented with a random subset of 6 texts from one of the 11 conditions: original, simplified by humans, or one of the nine automatic TS systems. Each passage-question pair is annotated by one native English speaker resulting in 1980 annotations in total. Annotations collected were manually spot-checked for straightlining (pattern where participants consistently select the same response option) and time differences to ensure that the participants were paying attention to the RC task.

7.2.2 Models for Evaluation

We generate simplified outputs for the selected passages at "Advanced" difficulty using the systems described below as they are representative of the variety of architectures and learning algorithms (supervised, unsupervised, black-box) proposed in the TS literature at the sentence or the paragraph levels. We also include the Elementary version of the text from the OneStopEnglish corpus to compare the reading comprehension of the original and model-generated simplified texts against a ground truth reference as a control condition.

a) Keep-it-simple (KIS) (Laban et al., 2021) is an unsupervised TS system trained using a reinforcement learning framework to enforce the generation of simple, adequate, and fluent outputs at the paragraph level.[2]

[2] https://github.com/tingofurro/keep_it_simple

b) MUSS (Martin et al., 2022) finetunes a BART-large (Lewis et al., 2020) model with control tokens (Martin et al., 2020b) extracted on paired text simplification datasets and/or mined paraphrases to train both supervised and unsupervised TS systems. [3] We use the suggested hyperparameters from the original paper to set the control tokens during simplification generation.[4]

c) ControlT5-Wiki (Sheang and Saggion, 2021) is a supervised controllable sentence simplification model that finetunes a T5-base model with control tokens. We set the control tokens to the suggested hyperparameters from the original paper.[5]

d) ControlSup (Scarton and Specia, 2018b) is a controllable supervised TS model that trains a transformer-based sequence-to-sequence model with U.S. target grade as a side-constraint to generate audience-specific simplified output. We generate simplified outputs corresponding to Grades 7 and 5 to match the target complexity of the human-written Elementary simplified texts and to assess the impact of the degree of simplification on correctness.

e) EditingCurriculum (EditCL), proposed by (Agrawal and Carpuat, 2022) trains a supervised edit-based non-autoregressive model that generates a simplified output for a desired target U.S. grade level through a sequence of edit operations like deletions and insertions applied to the complex input text. We generate simplified outputs corresponding to Grades 7 and 5.[6]

f) We generate paragraph-level simplified outputs using ChatGPT[7] with the following prompt:

[3] Please refer to Chapter 6 for more details.
[4] https://github.com/facebookresearch/muss
[5] https://github.com/KimChengSHEANG/TS_T5
[6] https://github.com/sweta20/EditingCL
[7] https://openai.com/blog/chatgpt

MODEL	TEXT			CONFIG		METRICS (P)		
	# (WORDS	# (SENTS)	FKGL	ARCH	DATA	SARI	BERTSCORE	
ORIGINAL	137.4	5.1	10.5	-	-	-	-	P
ELEMENTARY	118.1	5.7	7.4	-	-	-	-	P
MUSS-SUP	126.8	7.3	7.0	BART	WIKILARGE	45.07	0.940	S
CONTROLT5-WIKI	135.0	7.7	6.6	T5		44.76	0.938	S
CONTROLSUP-Grade7	132.8	5.9	9.0			29.27	0.946	S
CONTROLSUP-Grade5	124.1	7.3	6.8	TRANSFORMER	NEWSELA	38.35	0.939	S
EDITCL-Grade7	134.7	6.1	9.0			30.49	0.939	S
EDITCL-Grade5	131.0	8.3	6.1			39.69	0.929	S
CHATGPT	123.0	5.1	10.5	GPT-3.5	-	41.41	0.927	P
MUSS-UNSUP	124.7	5.7	9.1	BART	-	40.67	0.937	S
KIS	73.6	3.3	9.1	GPT-2	-	33.06	0.893	P

Table 7.1: Simplified texts are often shorter and include more sentences than the Original text. Automatic TS models use various architectures and datasets, generate simplified texts at either the sentence (S) or the paragraph (P) level, and show different tradeoffs in adequacy-simplicity (measured using BERTScore and SARI). Metrics are computed at the paragraph level (P). A 0.005 difference in BERTScore is significant (p-value = 0.00).

```
{Text}
Rewrite the above text so that it can be easily understood
by a non-native speaker of English:
```

Statistics for the human-written and automatically generated passages as well as model summary are presented in Table 7.1. Automatically generated or manually written simplified texts are shorter and include more sentences (due to sentence splitting) than the original unmodified text. Systems that use pre-trained knowledge (`MUSS, T5, ChatGPT`) receive a higher simplicity (SARI) score than models trained from scratch (`ControlSup, EditCL`) except `KIS` that achieves low simplicity and adequacy scores according to automatic metrics.[8] Both `ControlSup` and `EditCL` models generate simplified outputs at a higher complexity level than intended (Average FKGL for Grades 7 and 5 are Grades 9 and 7 respectively). Furthermore, the outputs span a wide range of adequacy

[8]SARI measures lexical simplification based on the words that are added, deleted, and kept by the systems by comparing system output against references and against the input sentence.

and simplicity scores where some systems trade-off adequacy for simplicity with low BERTScore and high SARI values (e.g. `ChatGPT, EditCL-Grade5`) and vice-versa (e.g. `ControlSup-Grade7`). While the range of BERTScore values appears small, differences of > 0.005 are statistically significant suggesting that the 0.4+ wide range includes meaningful differences within this set of systems.

7.3 Results

We first analyze the results to show the validity of this human evaluation set-up, before comparing the TS systems tested according to the correctness and answerability metrics.

7.3.1 Validity of the Human Evaluation

Results on human-written texts align with the literature. As can be seen in Table 7.3, human-written texts (`Original, Elementary`) achieve the highest correctness scores of approximately 78%. This aligns with results from Berzak et al. (2020) who report an accuracy of 80.7% with crowd workers from Prolific, when working over all 162 passages from OnestopQA. As expected, even with human written texts, participants do not answer all questions perfectly, reflecting individual differences in reading proficiency, background knowledge, and familiarity with the topic (Young, 1999; Rets and Rogaten, 2021) as well as the difficulty of the questions. These results thus provide an upper bound to contextualize the scores obtained by TS systems. Furthermore, the correctness scores obtained on the `Original` and `Elementary` versions of the text are very close, as expected when working with participants who are all native speakers.

Inter-annotator Agreement (IAA). We collect a second set of annotations for a subset of 6 articles for all the 11 systems (198) and compute the IAA using Cohen's kappa (McHugh,

2012) score. The IAA score for selecting the correct answer suggests a moderate agreement (0.437) despite the high subjectivity (individual comprehension differences) and complexity of the task (presence of 5 answer options).

Stability of Rankings Sampling 50 random subsets of k passages for each $k \in \{5, 10, 20, 30, 40, 50, 60\}$, we aggregate the mean correctness score for each subset size and show the rankings and variance for the systems in Figure 7.1 and Figure 7.2 respectively. Using > 40 unique passages for each system, i.e. approximately 120 questions stabilizes the rankings as well as reduces variance amongst the 11 systems, with `ChatGPT, T5` and `EditCL-Grade7, Grade5` system pairs achieving the same overall rank.

Taken together, these findings suggest that the evaluation framework is sound and provides a valid instrument to evaluate and compare systems.

7.3.2 Evaluating Correctness of Automatic TS outputs

TYPE	MODEL	PRE-TRAINED	% CORRECT	B	C	D	RANK	
HUMAN	ORIGINAL	-	78.33	6.11	2.22	1.11	**1**	
	ELEMENTARY	-	77.22	5.56	2.78	0.00	**2**	
SUPERVISED	MUSS-SUP	✓	76.11	6.67	1.67	1.67	**3**	
	CONTROLT5-WIKI	✓	74.44	6.11	2.78	1.67	**4**	$*$
	CONTROLSUP-Grade7	✗	70.56	3.89	2.78	2.78	**7**	
	EDITCL-Grade7	✗	69.44	10.56	2.22	0.56	**8**	$**$
	EDITCL-Grade5	✗	69.44	10.00	2.22	0.00	**8**	$**$
	CONTROLSUP-Grade5	✗	67.78	11.11	3.89	0.00	**10**	
BLACK BOX	CHATGPT	✓	74.44	9.44	1.11	0.00	**4**	$*$
UNSUPERVISED	MUSS-UNSUP	✓	73.33	6.67	2.78	1.11	**6**	
	KIS	✓	20.50	7.22	3.89	3.89	**11**	

Table 7.2: Results of the Correctness Evaluation: Supervised systems that leverage pre-trained knowledge achieve the highest accuracy on the RC tasks. $*$ and $**$: Systems that attain the same rank due to the same overall accuracy scores.

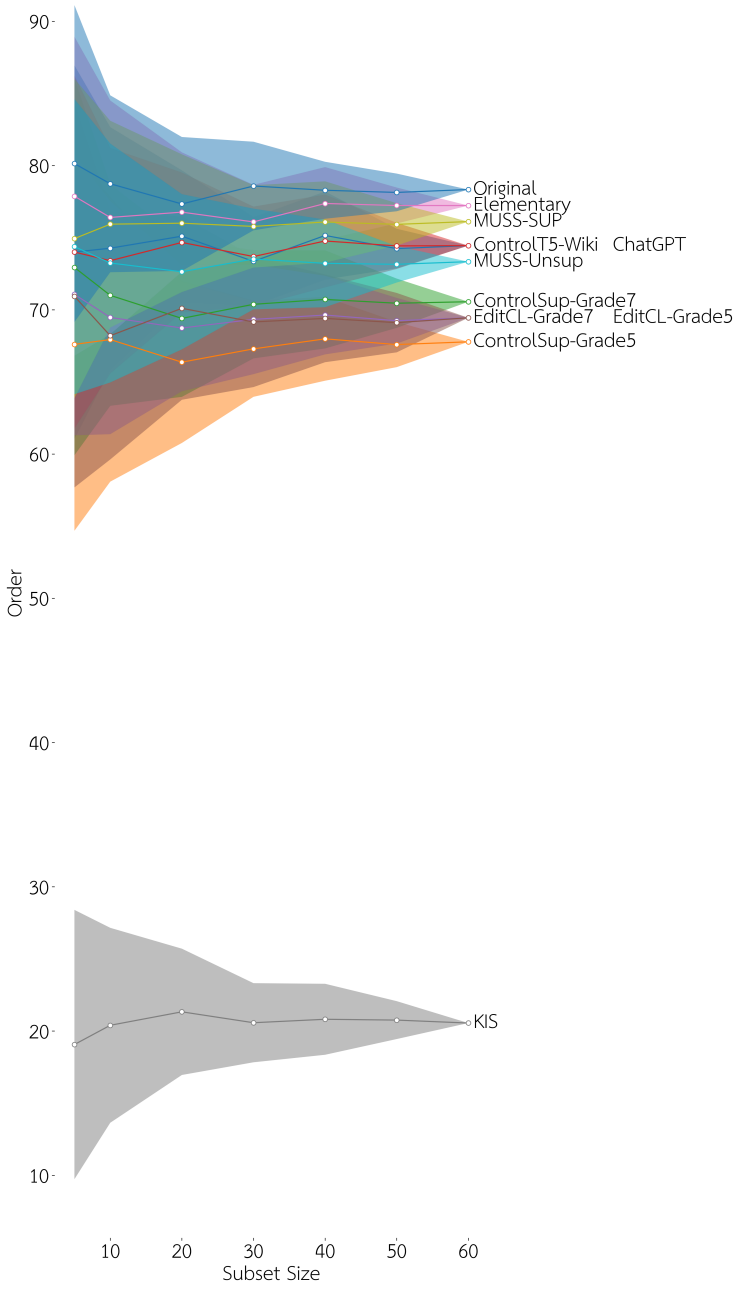


Figure 7.2: Variance reduces with an increase in the number of passages used to rank the systems as expected.

Table 7.2 shows the correctness (C) scores for human-written texts (`Original`, `Elementary`) and automatic simplifications generated from supervised (edit and non-

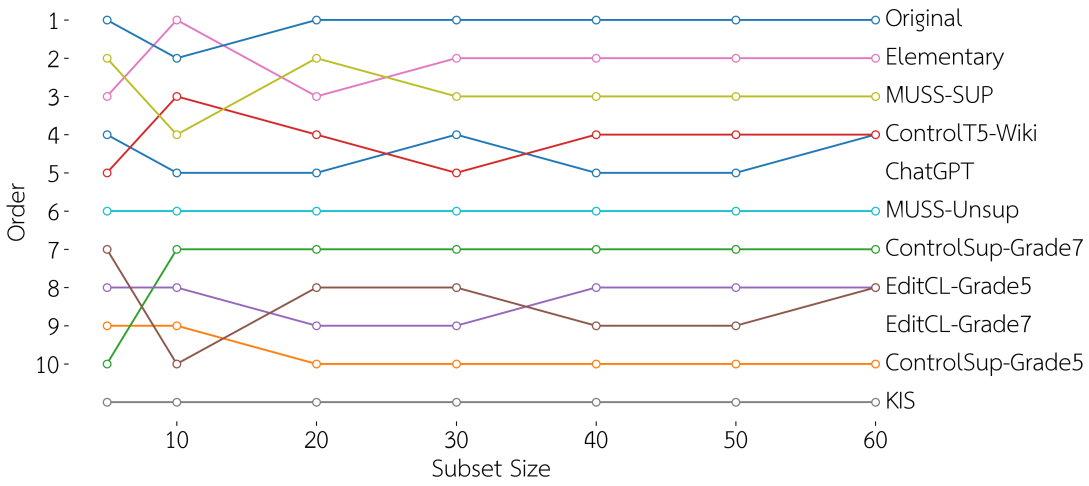


Figure 7.3: For every subset-size k, 50 runs are sampled and the % correct answers for each system are averaged. Rankings stabilize with a sample size of 40 unique articles.

edit based), unsupervised, and black-box LLMs. Systems achieve a wide range of scores, starting as low as 20% to approaching within 1% of the correctness achieved on human-written text. We discuss the correctness findings and compare systems in more detail below.

Results first show that unsupervised pre-training improves TS correctness. The significance of leveraging large-scale unsupervised pre-trained knowledge for modeling TS is demonstrated by MUSS-SUP, which achieves the highest overall accuracy among all the systems. Additionally, despite its unsupervised nature and the generation of simplified texts at the paragraph level, CHATGPT attains a similar correctness score to that of CONTROLT5-WIKI, a supervised sentence-level TS model, showing the benefits of large scale pre-training, and of reinforcement learning with human feedback – even though it is remains unfortunately unknown whether CHATGPT was trained on text simplification or related tasks. Overall, the scores show that the best performing TS systems rewrite content so that people understand the information tested as well as in human-written text. This suggests that those systems are above bar for moving forward with usability testing in future

work – thus asking not only whether rewrites are *correct* but whether they are useful to readers that need simplified text.

At the other end of the spectrum, the performance of KIS stands out with a remarkably low overall accuracy of only 20% on the generated simplified texts, attributable to hallucinated content and content deletion, as indicated by the low BERTScore in Table 7.1. We further validate this by analyzing the impact of content deletion on question answerability in the next section.

In the middle of the pack, among systems for grade-specific TS, edit-based models outperform autoregressive models. The autoregressive model ControlSup exhibits a 3% decrease in accuracy, due to a more aggressive deletion when simplifying to Grade 5, whereas edit-based models like EditCL maintain their correctness score even when generating simpler outputs at both grade levels 7 and 5 (Table 7.1). However, these edit-based models also result in miscomprehension as suggested by the relatively high percentage of questions marked with option B by the human participants. Note that Option B represents a plausible misunderstanding of the critical span upon which the question is based (Section 7.1). We hypothesize that this could be due to the reduced fluency of the model-generated simplifications via edit-based models.

7.3.3 TS Answerability Findings

We quantify the answerability score, Ans, for all automatic TS systems and human-written texts in Figure 7.6. As for correctness, human-written text do not yield perfect scores, thus providing context to interpret the scores of TS systems. Using the STARC annotation framework results in questions that should all in principle be answerable from the original text. Yet in practice participants still found 12% of questions to be unanswerable. Inspection shows that these questions require making complex inferences and hypotheses about the

plausibility of the various options. As a result, when given the new (E) answer option, they are more conservative in selecting the four other alternatives.

The unsupervised KIS stands out with the lowest A_S score of 35.56%, while the other systems range from 83-88%. Manual inspection shows that KIS outputs are particularly prone to hallucinations and over deletions. While the remaining systems achieve close scores, Figure 7.4 and 7.5, which display the questions correctly answered and marked with the UA option respectively for all passages and all conditions, show that no passage-question pairs are answered correctly by all models, suggesting that despite the close answerability scores, models do not make the same errors.

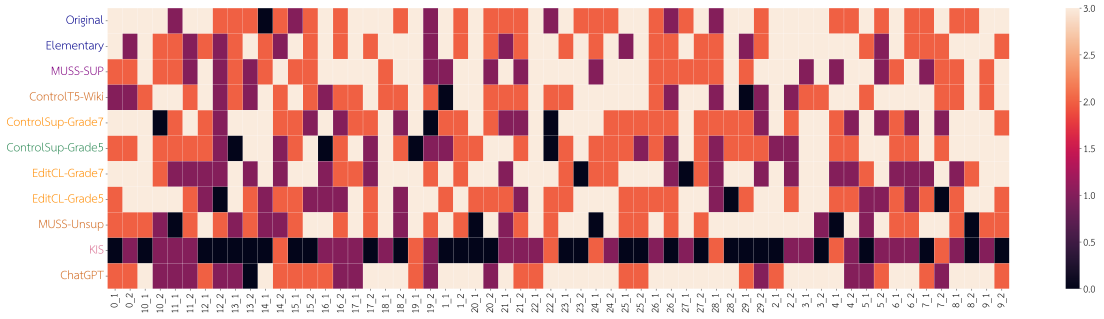


Figure 7.4: Number of correctly answered questions for each passage: None of the passages are correctly answered by all the models.

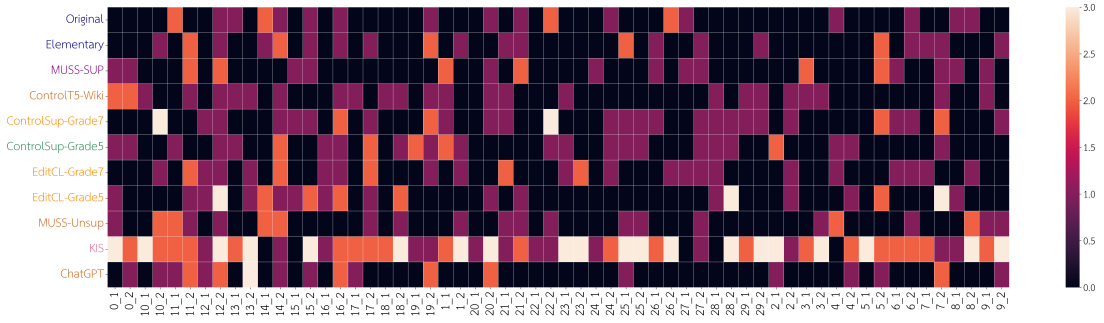


Figure 7.5: Number of non-answerable questions for each passage: Different models have different passage-questions pairs marked as non-answerable, suggesting different deletion errors.

Motivated by the manual inspection of KIS outputs, we hypothesize that over-deletion is the key culprit that makes questions unanswerable. To test this hypothesis automatically, we

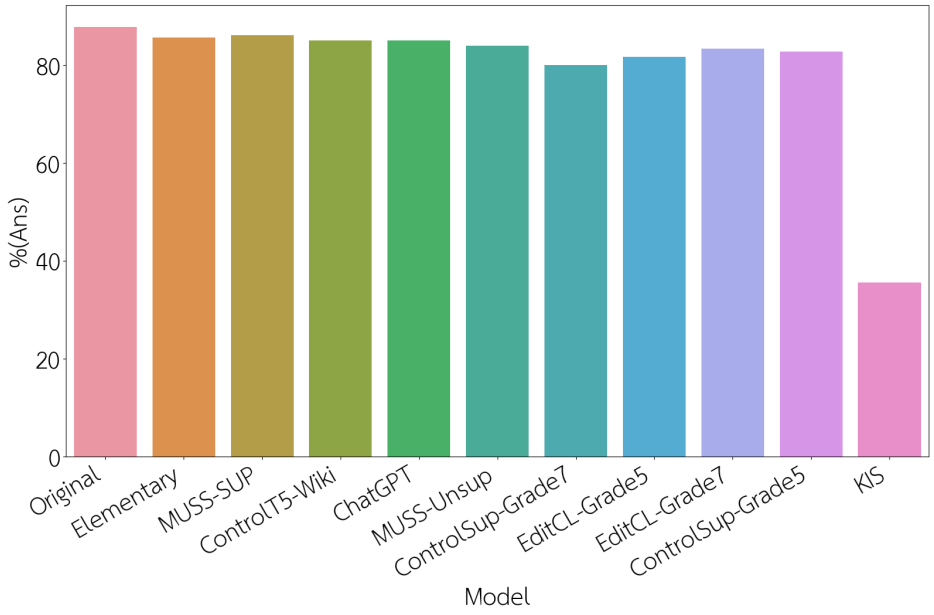


Figure 7.6: Participants mark 12-65% questions with UA, suggesting that the meaning of the original text is not entirely preserved in the simplified texts.

examine how the unigram overlap[9] between the question and the passage (`Support (Q)`) and the answer options and the passage (`Support (A)`) influence question answerability when model-generated outputs are used (Sugawara et al., 2018).

Figure 7.7 shows that `Support (A)`, is a more reliable predictor of answerability with a TPR: 0.675 at a FPR: 0.406 than `Support (Q)` (TPR: 0.565, FPR: 0.612). Answers that appear verbatim in the passage (Support(A)=1.0) are correctly answered 93% of the time. However, when the question lacks support in the passage, the unigram overlap with just the answer becomes an insufficient signal. Therefore, we also report the distribution of the product of `Support (A)` and `Support (Q)` in the same figure to directly capture the support for both the question and the answer options in the passage, i.e., the UA option. The resulting TPR rate for detecting unanswerable questions is 0.605 at an FPR of 0.353, indicating that the incorrect deletion of partial or complete phrases by the automatic TS systems affects both the support for the question and the answer options making RC question

[9]after stop word removal.

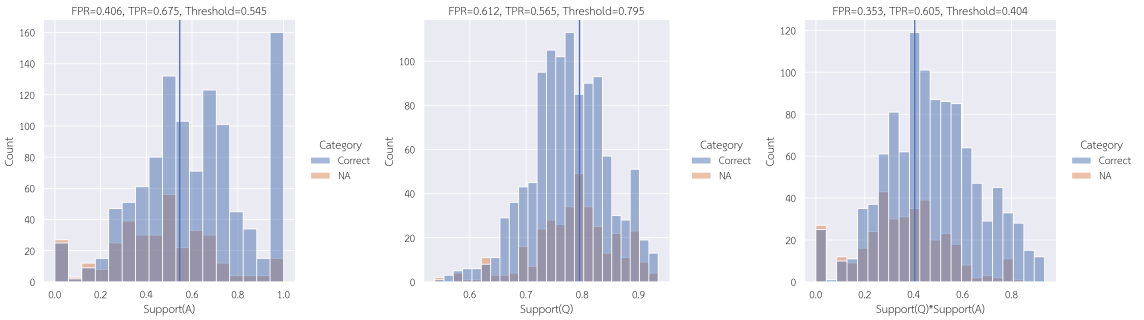


Figure 7.7: Unigram Overlap between the answer options and the passage (Support (A)), question and the passage (Support(Q)) and product of the two (Support (A)*Support (Q)) indicates content deletion as a major factor in making a question unanswerable. TPR: True Positive Rate; FPR: False Positive Rate.

unanswerable. These results temper the correctness results, suggesting that even the best-performing systems delete content, and thus motivating further human subject studies to develop machine-in-the-loop workflows to validate content before it is presented to readers. A relevant study conducted by Leroy et al. (2022) introduces an editor that generates word or phrase-level simplifications for complex texts. These suggestions are validated by a health educator to ensure they remain error-free before being presented to the end users. Future research can build upon this approach by developing editing tools for TS systems that operate at the sentence or paragraph-level.

7.4 Evaluating Automatic Evaluation Metrics

We now turn towards investigating to what extent automatic TS evaluation metrics frequently used in the literature capture the correctness system rankings obtained via reading comprehension (Al-Thanyyan and Azmi, 2021b; Maddela et al., 2022; Devaraj et al., 2022). We compute the Spearman-Rank correlation of the system-level scores using selected automatic metrics and % Correct from the RC task in Table 7.3. For meaning preservation, we evaluate BLEU, BERTScore (Zhang et al., 2020) and the Levenshtein distance computed between the system output with respect to the Elementary (REF) and the Original text (SRC)

respectively. For simplicity and readability dimensions, we report correlation scores with SARI, and FKGL respectively. Note that all metrics are computed at the paragraph level, just like in the reading comprehension task, unlike prior evaluation which uses and evaluates these metrics for sentence-level simplification.

METRICS	MEANING (REF)					MEANING (SRC)				SIMPLICITY	READABILITY
	BLEU	BERTSCORE			LEVDIST	BERTSCORE			LEVDIST	SARI	FKGL
		(P)	(R)	(F1)		(P)	(R)	(F1)			
ALL	-0.193	0.418	0.292	0.310	-0.142	0.084	0.033	0.033	-0.159	**0.728**	0.126
ALL-{KIS}	0.157	0.167	-0.012	0.012	-0.634	-0.311	-0.383	-0.383	-0.659	**0.778**	0.335

Table 7.3: Evaluation of Automatic Metrics computed at the Paragraph-level using All (9) and All but KIS (8) automatic TS systems: SARI achieves the highest correlation with human judgments, followed by LevDist, surpassing meaning preservation and readability metrics: BLEU, BERTScore, and FKGL.

Overall, SARI achieves the best correlation across the board even after removing the outlier system, i.e. `KIS`. The Levenshtein edit distance of the system output with the `Original` (-0.659) and the `Elementary` (-0.634) text receives a negative moderate-high correlation with human judgments, outperforming both surface-form (BLEU) and embedding-based metric (BERTScore) after removing the outlier system (KIS). We hypothesize that the 3-way comparison between the input, the output, and the reference as computed by SARI is key in yielding system rankings that are consistent with those based on our correctness results, yet could be further repurposed to more directly match correctness scores.

7.5 Model-based Question Answering

Our evaluation so far has relied on human-written questions answered by crowd workers, using either model-generated or human-written texts. Automating one or both components would help scale the evaluation and port it to new settings more flexibly. Recent work suggests that this might be plausible: Krubiński et al. (2021) show that automatically

generated questions and answers can be used to evaluate Machine Translation systems at the sentence level, and automatic QA techniques (Fabbri et al., 2022; Wang et al., 2020) have been used to assess the factuality of summarization systems.

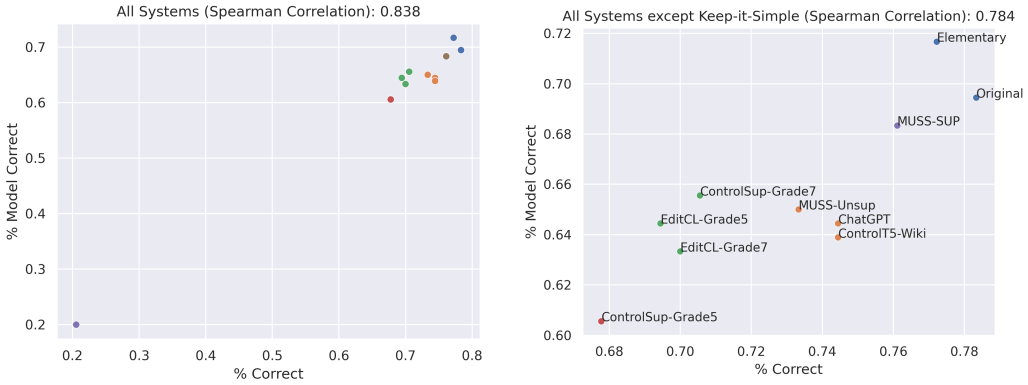


Figure 7.8: Model-based QA using `UnifiedQA v2` achieves moderate-high correlation with human judgments but fails to distinguish closely competing systems.

Here, we assess the performance of a state-of-the-art QA system in recovering the gold-standard correctness ranking induced by human judgments, leaving the more complex study of multiple-choice RC question generation to future work. We use `UnifiedQA v2` [10] a QA model, trained to answer questions in 4 different formats using 20 different datasets. This model has been shown to support better generalization to unseen datasets compared to models specialized for individual datasets. We use the format recommended in the original paper: `{question} \n (A) {choice 1} (B) {choice 2} ... \n {paragraph}` to generate answers for all the conditions. The correlation between Exact Match (EM) accuracy for each system and the ground truth accuracy (% Correct) is shown in Figure 7.8. While the QA model achieves a Spearman-rank correlation of 0.744 (excluding the outlier system: KIS), its ability to distinguish closely competing systems (highlighted by the same color in the graph) is limited. Interestingly, model-based QA using human-simplified text achieves higher accuracy than using original unmodified text. This

[10] `allenai/unifiedqa-v2-t5-3b-1363200`

finding is in line with prior work where TS has been shown to improve the performance of multiple downstream NLP tasks (Section 2.1.2). This suggests that automating part of the evaluation framework is a direction worth investigating in more depth in future work.

7.6 Conclusion

In this chapter, we introduce an evaluation framework based on reading comprehension to assess the correctness of simplified texts. We conduct a thorough human evaluation of 10 simplified texts (one human-written and nine by automatic TS systems) using our proposed framework. Supervised systems that leverage pre-training knowledge (MUSS, T5) achieve the highest accuracy on the RC tasks among the automatic controllable TS systems. Our results show that edit-based models can generate equally correct outputs at different complexity levels compared to supervised autoregressive models. However, we also find that systems do not preserve the meaning of the original text leading to at least 14% of the questions being marked as "unanswerable" even by the best-performing supervised system. A meta-evaluation of existing automatic metrics frequently used in TS evaluation indicates that the 3-way comparison used in SARI makes it a reliable metric for system-level evaluation at the paragraph level. Furthermore, while model-based QA can approximate system rankings induced by human evaluation, we caution that the rankings of the system might be influenced by the impact of TS on the QA task.

Overall, these results confirm the importance of directly evaluating the correctness of the information conveyed by TS systems. The framework also provides a blueprint to evaluate whether correct TS outputs improve reading comprehension for people who have difficulty understanding complex texts, which we intend to investigate in future work.

Chapter 8: Conclusion and Future Work

8.1 Summary

In this dissertation, we introduced tasks, datasets, models, and algorithms that let users specify how simple or complex the generated text should be in a given language.

First, we empirically explored the capabilities of existing MT systems in capturing the diversity of translations generated by humans and showed that while standard MT systems can generate multiple translations to some extent, they fail to capture the full spectrum of the variety of possible valid translations generated by humans (Agrawal and Carpuat, 2020).

In order to capture the target audience more directly and to systematically navigate the space of possible translations, we introduced the Complexity Controlled Machine Translation task that lets the user specify the desired complexity level (as measured by the U.S. reading level) in the generated text when translating text from one language to another. We contributed datasets and models that can enable the generation and evaluation of such outputs (Agrawal and Carpuat, 2019).

Furthermore, recognizing that when humans simplify a complex text in a given language, they often revise parts of the complex text according to the intended audience, we presented strategies to adopt general-purpose Edit-based Non-Autoregressive models for controllable text simplification. Our proposed inference (Agrawal et al., 2021b) and training (Agrawal and Carpuat, 2022) algorithms inject additional guidance for improving the edit-based

model's ability to learn to perform a wide range of simplification-specific edit operations for different grade levels.

We then presented an empirical study that analyzed how the nature and the type of control introduced in sequence-to-sequence models as side constraints impact the degree and nature of simplification performed. These methods raised a practical challenge: there is no systematic method to set these control tokens at inference time. Hence, we proposed simple data-driven solutions to adapt controllable TS models that leverage unsupervised pre-training and control tokens that describe the nature of simplification operations for grade-specific TS. We showed that our approach improves the simplicity of the outputs and resulted in better coverage of the range of edit operations found in human-written simplifications.

Finally, we designed a new evaluation framework and conduct a thorough human evaluation for the correctness of texts generated by several controllable TS models. Supervised systems which utilize pre-training knowledge exhibited the highest accuracy in answering reading comprehension (RC) tasks among automatic controllable TS systems. However, a notable drawback was that these systems sometimes failed to maintain the original meaning of the text, resulting in at least 14% of the RC questions being deemed "unanswerable" even for the top-performing supervised system. This suggests that even the best-performing models are not ready for deployment as is, and that human editors are needed to validate and post-edit the outputs before they are presented to the intended readers. Furthermore, while our evaluation primarily focused on the correctness of the generated simplified texts, it still remains an open question on whether the nature of simplifications generated by these systems, when *correct*, are useful for the target audiences.

8.2 Limitations and Future Work

In this section, we discuss some of the limitations of our work and point out directions for future research. We acknowledge that NLG systems are continuously advancing, and the methods to tackle the gaps we have addressed in the thesis and described below may evolve. Nevertheless, this dissertation contributes problem framings and datasets that we are motivated by real needs and we hope will thus remain relevant no matter which forms the underlying models and technical solutions might take in the future. Furthermore, our study of complexity controlled language generation calls for future work on accommodating individual user preferences in broader settings (including domains and languages), as well as on designing reliable and useful indicators of output quality and uncertainty.

8.2.1 Complexity Controlled Text Generation for Catering Individual Preferences Beyond English

In this thesis, we focus on controlling the complexity (as specified by the U.S. reading grade level) of text in two natural language generation tasks — monolingual text simplification and complexity-controlled machine translation. Given this information, we present several strategies to generate text in the target language, English that matches the desired complexity level. However, the individual needs of different users might still differ based on their knowledge, background, and native language. For example, people with dyslexia are highly individual in what material they deem easy and complex (Ziegler et al., 2008). Ideally, in order to be able to cater to the needs of individual user preferences across different languages, it might be desirable to design and provide more *interactive* control to the users over what gets simplified or to be specified by a teacher. Striking the right balance between

simplicity and fidelity to the source would require continuous refinement of these systems based on continuous user feedback.

The need to preserve information in simplified text for specialized fields such as medicine or law poses challenges in the design of content-preserving simplification systems. We primarily evaluated our systems on datasets from the news domain with simplifications targeted toward students at various grade levels. However, depending on the target domain, it might be desirable to preserve certain domain-specific terminologies in the simplified text. Designing content-preserving simplification systems targeted toward specific domains' content might require employing harder constraints or elaborative text simplification mechanisms which we leave to future work.

Designing better systems for languages other than English will also require investing in language-specific evaluation design. While there has been a lot of work in designing readability metrics for English which in turn facilitates automatic evaluation, it remains unclear how the representation of complexity should be defined in other languages and what aspects of text complexity might transfer across languages.

8.2.2 Leveraging Large-Scale Models for Controllable Text Generation

Recent large-scale pre-trained language models like ChatGPT have shown remarkable abilities in many natural language generation tasks and are able to produce coherent and contextually relevant text. While they are powerful, controlling their output quality and diversity is challenging. Recent works have employed prompting techniques to control attributes of machine-generated translations using exemplars like formality (Garcia et al., 2023) or domain (Agrawal et al., 2022b). These pre-trained LMs also exhibit exceptional abilities to follow editing instructions for tasks like Grammatical Error Correction (Loem et al., 2023). Our human evaluation (Chapter 7) shows that ChatGPT while not trained with

known explicit learning signals for text simplification is still able to achieve competitive performance with supervised systems. However, the application of these LMs in controlling multiple attributes of text simultaneously (for example: translating and simplifying) remains unexplored. Tailoring these LMs to control specific properties of the generated text in the target language would require designing techniques that can perform local and global editing of the original content which we leave to future work.

8.2.3 Designing Quality and Uncertainty Indicators for Useful and Reliable Controllable NLG Systems

Ensuring the quality and correctness of the automatically generated text is crucial to maintain user trust and satisfaction. Deploying text generation systems for real-world applications will require making end users aware of when and how these systems fail and designing interpretable quality control indicators. NLG models often lack the ability to convey the level of confidence or uncertainty associated with their generated output due to miscalibrated probability distribution and a large search space (Lemon et al., 2010; Xiao and Wang, 2021; Kuhn et al., 2022; Lin et al., 2022). Developing uncertainty indicators that enable NLG systems to express degrees of confidence, identify areas of uncertainty, detect when a model-generated text is usable, and provide appropriate caveats or explanations can greatly enhance the transparency and trustworthiness of the generated content. These issues have received significantly more attention in the context of machine translation (Ott et al., 2018; Glushkova et al., 2021; Gladkoff et al., 2022), including some work of our own (Agrawal et al., 2021a, 2022a; Xu et al., 2023). Future work could draw inspiration from MT findings to develop quality feedback mechanisms for a broader range of generation tasks.

8.3 Ethical Considerations in Deploying NLG Systems in the Real World

While NLG systems are designed with the intention of assisting users in understanding content better, the potential errors introduced by these systems and ambiguous interpretations of the generated text can pose risks and harm to end users. Recognizing and acknowledging that risks can come with all generation systems (Bender et al., 2021; Bommasani et al., 2021; Weidinger et al., 2022), we highlight some issues that are particularly salient in text simplification systems. For example, the very nature of simplification involves content removal and rephrasing complex concepts, which can sometimes result in oversimplification or loss of critical nuances as evidenced in prior work (Devaraj et al., 2022) and our human evaluation (Chapter 7). As a consequence, users relying solely on simplified texts may develop an incomplete or inaccurate understanding of the subject matter, leading to potential misconceptions or misinterpretations. Therefore, users should be informed about the limitations of the deployed model and the types of errors that can occur. One way to achieve this is by providing quality indicators that highlight the differences between the generated text and the original content, thus better-equipping users to understand the nature of the generated text.

Furthermore, the model's responses are based on the patterns and information it learned from the training data, which may not always align with the specific requirements or constraints provided by the user (Paullada et al., 2021). The ever-improving ability of these NLG models to generate human-like text also makes them susceptible to further amplifying the biases observed in the training data. Hence, it is important to ensure that the deployment of such systems does not create a digital divide or exclude marginalized populations. For example, it is essential to incorporate diverse and inclusive datasets that encompass a wide range of demographics, including marginalized populations, when training NLG models

(Rogers, 2021). Also, rigorous testing and evaluation procedures should be implemented to identify and rectify biases detected within the NLG systems.

One potential way moving forward is to take a more human-centered approach to developing and designing complexity-controlled text generation systems that can account for the needs of the many stakeholders (including readers, editors, teachers, and developers). This would require engaging with the users throughout the development and deployment process so that the collaborative process can ensure that the NLG systems are tailored to meet the diverse needs of its users.

Bibliography

Eneko Agirre, Carmen Banea, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Rada Mihalcea, German Rigau, and Janyce Wiebe. 2016. SemEval-2016 task 1: Semantic textual similarity, monolingual and cross-lingual evaluation. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 497–511, San Diego, California. Association for Computational Linguistics.

Sweta Agrawal and Marine Carpuat. 2019. Controlling text complexity in neural machine translation. 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 1549–1564, Hong Kong, China. Association for Computational Linguistics.

Sweta Agrawal and Marine Carpuat. 2020. Generating diverse translations via weighted fine-tuning and hypotheses filtering for the Duolingo STAPLE task. In *Proceedings of the Fourth Workshop on Neural Generation and Translation*, pages 178–187, Online. Association for Computational Linguistics.

Sweta Agrawal and Marine Carpuat. 2022. An imitation learning curriculum for text editing with non-autoregressive models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7550–7563, Dublin, Ireland. Association for Computational Linguistics.

Sweta Agrawal, George Foster, Markus Freitag, and Colin Cherry. 2021a. Assessing reference-free peer evaluation for machine translation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1158–1171, Online. Association for Computational Linguistics.

Sweta Agrawal, Nikita Mehandru, Niloufar Salehi, and Marine Carpuat. 2022a. Quality estimation via backtranslation at the WMT 2022 quality estimation task. In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 593–596, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Sweta Agrawal, Weijia Xu, and Marine Carpuat. 2021b. A non-autoregressive edit-based approach to controllable text simplification. In *Findings of the Association for Com-*

putational Linguistics: ACL-IJCNLP 2021, pages 3757–3769, Online. Association for Computational Linguistics.

Sweta Agrawal, Chunting Zhou, Mike Lewis, Luke Zettlemoyer, and Marjan Ghazvininejad. 2022b. In-context examples selection for machine translation. *arXiv preprint arXiv:2212.02437*.

Suha S Al-Thanyyan and Aqil M Azmi. 2021a. Automated text simplification: A survey. *ACM Computing Surveys (CSUR)*, 54(2):1–36.

Suha S. Al-Thanyyan and Aqil M. Azmi. 2021b. Automated text simplification: A survey. *ACM Comput. Surv.*, 54(2).

David Allen. 2009. A study of the role of relative clauses in the simplification of news texts for learners of english. *System*, 37(4):585–599.

Oliver Alonzo, Matthew Seita, Abraham Glasser, and Matt Huenerfauth. 2020. Automatic text simplification tools for deaf and hard of hearing adults: Benefits of lexical simplification and providing users with autonomy. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI '20, page 1–13, New York, NY, USA. Association for Computing Machinery.

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.

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.

Fernando Emilio Alva-Manchego, Carolina Scarton, and Lucia Specia. 2020. Data-driven sentence simplification: Survey and benchmark. *Comput. Linguistics*, 46(1):135–187.

Mandya Angrosh, Tadashi Nomoto, and Advaith Siddharthan. 2014. Lexico-syntactic text simplification and compression with typed dependencies. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 1996–2006, Dublin, Ireland. Dublin City University and Association for Computational Linguistics.

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 4251–4261.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural Machine Translation by Jointly Learning to Align and Translate. In *International Conference on Learning Representations (ICLR)*.

Satanjeev Banerjee and Alon Lavie. 2005. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72.

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.

Luisa Bentivogli, Beatrice Savoldi, Matteo Negri, Mattia A. Di Gangi, Roldano Cattoni, and Marco Turchi. 2020. Gender in danger? evaluating speech translation technology on the MuST-SHE corpus. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6923–6933, Online. Association for Computational Linguistics.

Jan Berka, Ondrej Bojar, et al. 2011. Quiz-based evaluation of machine translation. *The Prague Bulletin of Mathematical Linguistics*, 95:77.

Yevgeni Berzak, Jonathan Malmaud, and Roger Levy. 2020. STARC: Structured annotations for reading comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5726–5735, Online. Association for Computational Linguistics.

Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. 2021. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*.

Kiante Brantley, Kyunghyun Cho, Hal Daumé, and Sean Welleck. 2019. Non-monotonic sequential text generation. In *Proceedings of the 2019 Workshop on Widening NLP*, pages 57–59, Florence, Italy. Association for Computational Linguistics.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Yvonne Canning, John Tait, Jackie Archibald, and Ros Crawley. 2000. Cohesive generation of syntactically simplified newspaper text. In *International Workshop on Text, Speech and Dialogue*, pages 145–150. Springer.

John Carroll, Guido Minnen, Darren Pearce, Yvonne Canning, Siobhan Devlin, and John Tait. 1999. Simplifying text for language-impaired readers. In *Ninth Conference of the European Chapter of the Association for Computational Linguistics*.

William Chan, Chitwan Saharia, Geoffrey Hinton, Mohammad Norouzi, and Navdeep Jaitly. 2020. Imputer: Sequence modelling via imputation and dynamic programming. In *International Conference on Machine Learning*, pages 1403–1413. PMLR.

R. Chandrasekar, Christine Doran, and B. Srinivas. 1996a. Motivations and Methods for Text Simplification. In *Proceedings of the 16th Conference on Computational Linguistics - Volume 2*, COLING '96, pages 1041–1044, Stroudsburg, PA, USA. Association for Computational Linguistics.

R. Chandrasekar, Christine Doran, and B. Srinivas. 1996b. Motivations and methods for text simplification. In *COLING 1996 Volume 2: The 16th International Conference on Computational Linguistics*.

Mia Xu Chen, Orhan Firat, Ankur Bapna, Melvin Johnson, Wolfgang Macherey, George Foster, Llion Jones, Niki Parmar, Mike Schuster, Zhifeng Chen, et al. 2018. The best of both worlds: Combining recent advances in neural machine translation. *arXiv preprint arXiv:1804.09849*.

Colin Cherry and George Foster. 2012. Batch tuning strategies for statistical machine translation. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 427–436. Association for Computational Linguistics.

Kenneth Ward Church and Eduard H. Hovy. 2004. Good applications for crummy machine translation. *Machine Translation*, 8:239–258.

William Coster and David Kauchak. 2011a. Simple English Wikipedia: A New Text Simplification Task. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: Short Papers - Volume 2*, HLT '11, pages 665–669, Stroudsburg, PA, USA. Association for Computational Linguistics.

William Coster and David Kauchak. 2011b. Simple English Wikipedia: A new text simplification task. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 665–669, Portland, Oregon, USA. Association for Computational Linguistics.

Scott A Crossley, Hae Sung Yang, and Danielle S McNamara. 2014. What's so simple about simplified texts? a computational and psycholinguistic investigation of text comprehension and text processing. *Reading in a Foreign Language*, 26(1):92–113.

Sumanth Dathathri, Andrea Madotto, Janice Lan, Jane Hung, Eric Frank, Piero Molino, Jason Yosinski, and Rosanne Liu. 2019. Plug and play language models: A simple approach to controlled text generation. In *International Conference on Learning Representations*.

Ashwin Devaraj, William Sheffield, Byron Wallace, and Junyi Jessy Li. 2022. Evaluating factuality in text simplification. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7331–7345, Dublin, Ireland. 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.

Jennifer B. Doyon, Kathryn B. Taylor, and John S. White. 1999. Task-based evaluation for machine translation. In *Proceedings of Machine Translation Summit VII*, pages 574–578, Singapore, Singapore.

Sergey Edunov, Myle Ott, Michael Auli, David Grangier, and Marc'Aurelio Ranzato. 2018. Classical structured prediction losses for sequence to sequence learning. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 355–364, New Orleans, Louisiana. Association for Computational Linguistics.

Noemie Elhadad and Komal Sutaria. 2007. Mining a lexicon of technical terms and lay equivalents. In *Proceedings of the Workshop on BioNLP 2007: Biological, Translational, and Clinical Language Processing*, pages 49–56. Association for Computational Linguistics.

Alexander Fabbri, Chien-Sheng Wu, Wenhao Liu, and Caiming Xiong. 2022. QAFactEval: Improved QA-based factual consistency evaluation for summarization. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2587–2601, Seattle, United States. Association for Computational Linguistics.

Angela Fan, Mike Lewis, and Yann Dauphin. 2018. Hierarchical neural story generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 889–898, Melbourne, Australia. Association for Computational Linguistics.

Marcello Federico and Mauro Cettolo. 2007. Efficient handling of n-gram language models for statistical machine translation. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 88–95, Prague, Czech Republic. Association for Computational Linguistics.

M. Fomicheva, Shuo Sun, L. Yankovskaya, F. Blain, Francisco Guzmán, M. Fishel, Nikolaos Aletras, Vishrav Chaudhary, and Lucia Specia. 2020. Unsupervised quality estimation for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:539–555.

Mikel L. Forcada, Carolina Scarton, Lucia Specia, Barry Haddow, and Alexandra Birch. 2018. Exploring gap filling as a cheaper alternative to reading comprehension questionnaires when evaluating machine translation for gisting. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 192–203, Brussels, Belgium. Association for Computational Linguistics.

Markus Freitag, David Grangier, and Isaac Caswell. 2020. BLEU might be guilty but references are not innocent. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 61–71, Online. Association for Computational Linguistics.

Núria Gala, Anaïs Tack, Ludivine Javourey-Drevet, Thomas François, and Johannes C. Ziegler. 2020. Alector: A parallel corpus of simplified French texts with alignments of misreadings by poor and dyslexic readers. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 1353–1361, Marseille, France. European Language Resources Association.

William A Gale and Kenneth W Church. 1993. A program for aligning sentences in bilingual corpora. *Computational linguistics*, 19(1):75–102.

William A. Gale, Kenneth Ward Church, et al. 1994. A program for aligning sentences in bilingual corpora. *Computational linguistics*, 19(1):75–102.

Cristina Garbacea, Mengtian Guo, Samuel Carton, and Qiaozhu Mei. 2021. Explainable prediction of text complexity: The missing preliminaries for text simplification. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1086–1097, Online. Association for Computational Linguistics.

Xavier Garcia, Yamini Bansal, Colin Cherry, George Foster, Maxim Krikun, Fangxiaoyu Feng, Melvin Johnson, and Orhan Firat. 2023. The unreasonable effectiveness of few-shot learning for machine translation. *arXiv preprint arXiv:2302.01398*.

Hu Gengshen. 2003. Translation as adaptation and selection. *Perspectives: Studies in Translatology*, 11(4):283–291.

Laurie Gerber and Eduard Hovy. 1998. Improving Translation Quality by Manipulating Sentence Length. In *Conference of the Association for Machine Translation in the Americas (AMTA)*.

Marjan Ghazvininejad, Omer Levy, Yinhan Liu, and Luke Zettlemoyer. 2019. Mask-predict: Parallel decoding of conditional masked language models. 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 6114–6123.

Marjan Ghazvininejad, Xing Shi, Jay Priyadarshi, and Kevin Knight. 2017. Hafez: an interactive poetry generation system. In *Proceedings of ACL 2017, System Demonstrations*, pages 43–48, Vancouver, Canada. Association for Computational Linguistics.

Serge Gladkoff, Irina Sorokina, Lifeng Han, and Alexandra Alekseeva. 2022. Measuring uncertainty in translation quality evaluation (TQE). In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 1454–1461, Marseille, France. European Language Resources Association.

Taisiya Glushkova, Chrysoula Zerva, Ricardo Rei, and André F. T. Martins. 2021. Uncertainty-aware machine translation evaluation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3920–3938, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Sian Gooding, Yevgeni Berzak, Tony Mak, and Matt Sharifi. 2021. Predicting text readability from scrolling interactions. In *Proceedings of the 25th Conference on Computational Natural Language Learning*, pages 380–390, Online. Association for Computational Linguistics.

Yvette Graham, Timothy Baldwin, Meghan Dowling, Maria Eskevich, Teresa Lynn, and Lamia Tounsi. 2016. Is all that glitters in machine translation quality estimation really gold? In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 3124–3134, Osaka, Japan. The COLING 2016 Organizing Committee.

Alex Graves. 2012. Sequence transduction with recurrent neural networks. *arXiv preprint arXiv:1211.3711*.

Jiatao Gu, James Bradbury, Caiming Xiong, Victor O.K. Li, and Richard Socher. 2018. Non-autoregressive neural machine translation. In *International Conference on Learning Representations*.

Jiatao Gu, Changhan Wang, and Jake Zhao. 2019. Levenshtein transformer. In *NeurIPS*.

Junliang Guo, Xu Tan, Linli Xu, Tao Qin, Enhong Chen, and Tie-Yan Liu. 2020. Fine-tuning by curriculum learning for non-autoregressive neural machine translation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 7839–7846.

Eva Hasler, Adrià de Gispert, Felix Stahlberg, Aurelien Waite, and Bill Byrne. 2017. Source sentence simplification for statistical machine translation. *Computer Speech & Language*, 45(Supplement C):221–235.

Hany Hassan, Anthony Aue, Chang Chen, Vishal Chowdhary, Jonathan Clark, Christian Federmann, Xuedong Huang, Marcin Junczys-Dowmunt, William Lewis, Mu Li, Shujie Liu, Tie-Yan Liu, Renqian Luo, Arul Menezes, Tao Qin, Frank Seide, Xu Tan, Fei Tian, Lijun Wu, Shuangzhi Wu, Yingce Xia, Dongdong Zhang, Zhirui Zhang, and Ming Zhou. 2018. Achieving human parity on automatic chinese to english news translation. *CoRR*, abs/1803.05567.

He He, Jason Eisner, and Hal Daume. 2012. Imitation learning by coaching. *Advances in Neural Information Processing Systems*, 25:3149–3157.

Kenneth Heafield. 2011. KenLM: Faster and smaller language model queries. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 187–197, Edinburgh, Scotland. Association for Computational Linguistics.

Michael Heilman, Aoife Cahill, Nitin Madnani, Melissa Lopez, Matthew Mulholland, and Joel Tetreault. 2014. Predicting grammaticality on an ordinal scale. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 174–180.

Michael Heilman, Kevyn Collins-Thompson, and Maxine Eskenazi. 2008. An analysis of statistical models and features for reading difficulty prediction. In *Proceedings of the third workshop on innovative use of NLP for building educational applications*, pages 71–79. Association for Computational Linguistics.

Maryam Heydari, Morteza Khodabandehlou, and Shahrokh Jahandar. 2013. On the effectiveness of strategy-based instruction of textual simplification on efl learners' reading comprehension ability. *Indian Journal of Fundamental and Applied Life Sciences*, 3(2):176–183.

Felix Hieber, Tobias Domhan, Michael Denkowski, David Vilar, Artem Sokolov, Ann Clifton, and Matt Post. 2017. Sockeye: A toolkit for neural machine translation. *arXiv preprint arXiv:1712.05690*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Chris Hokamp and Qun Liu. 2017. Lexically constrained decoding for sequence generation using grid beam search. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1535–1546, Vancouver, Canada. Association for Computational Linguistics.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration. In *International Conference on Learning Representations*.

Ari Holtzman, Jan Buys, Maxwell Forbes, Antoine Bosselut, David Golub, and Yejin Choi. 2018. Learning to write with cooperative discriminators. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1638–1649, Melbourne, Australia. Association for Computational Linguistics.

Eduard Hovy. 1987. Generating natural language under pragmatic constraints. *Journal of Pragmatics*, 11(6):689–719.

Eduard H Hovy. 1990. Pragmatics and natural language generation. *Artificial Intelligence*, 43(2):153–197.

Cheng-Yu Hsieh, Yi-An Lin, and Hsuan-Tien Lin. 2018. A deep model with local surrogate loss for general cost-sensitive multi-label learning. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Kentaro Inui, Atsushi Fujita, Tetsuro Takahashi, Ryu Iida, and Tomoya Iwakura. 2003. Text simplification for reading assistance: A project note. In *Proceedings of the Second International Workshop on Paraphrasing*, pages 9–16, Sapporo, Japan. Association for Computational Linguistics.

Chao Jiang, Mounica Maddela, Wuwei Lan, Yang Zhong, and Wei Xu. 2020. Neural crf model for sentence alignment in text simplification. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*.

Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's Multilingual Neural Machine Translation System: Enabling Zero-Shot Translation. *arXiv:1611.04558 [cs]*.

Douglas Jones, Edward Gibson, Wade Shen, Neil Granoien, Martha Herzog, Douglas Reynolds, and Clifford Weinstein. 2005a. Measuring human readability of machine generated text: three case studies in speech recognition and machine translation. In *Proceedings.(ICASSP'05). IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005.*, volume 5, pages v–1009. IEEE.

Douglas Jones, Wade Shen, Neil Granoien, Martha Herzog, and Clifford Weinstein. 2005b. Measuring translation quality by testing english speakers with a new defense language proficiency test for arabic. Technical report, MASSACHUSETTS INST OF TECH LEXINGTON LINCOLN LAB.

Tomoyuki Kajiwara. 2019. Negative lexically constrained decoding for paraphrase generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6047–6052, Florence, Italy. Association for Computational Linguistics.

Nitish Shirish Keskar, Bryan McCann, Lav R Varshney, Caiming Xiong, and Richard Socher. 2019. Ctrl: A conditional transformer language model for controllable generation. *arXiv preprint arXiv:1909.05858*.

Tannon Kew and Sarah Ebling. 2022. Target-level sentence simplification as controlled paraphrasing. In *Proceedings of the Workshop on Text Simplification, Accessibility, and Readability (TSAR-2022)*, pages 28–42, Abu Dhabi, United Arab Emirates (Virtual). Association for Computational Linguistics.

Yuta Kikuchi, Graham Neubig, Ryohei Sasano, Hiroya Takamura, and Manabu Okumura. 2016. Controlling output length in neural encoder-decoders. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1328–1338.

Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *International Conference on Learning Representations*.

Sigrid Klerke, Sheila Castilho, Maria Barrett, and Anders Søgaard. 2015. Reading metrics for estimating task efficiency with MT output. In *Proceedings of the Sixth Workshop on Cognitive Aspects of Computational Language Learning*, pages 6–13, Lisbon, Portugal. Association for Computational Linguistics.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open Source Toolkit for Statistical Machine Translation. In *Annual Meeting of the Association for Computational Linguistics (ACL), Demonstration Session*, Prague, Czech Republic.

Philipp Koehn and Rebecca Knowles. 2017. Six challenges for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*, pages 28–39, Vancouver. Association for Computational Linguistics.

Reno Kriz, João Sedoc, Marianna Apidianaki, Carolina Zheng, Gaurav Kumar, Eleni Miltsakaki, and Chris Callison-Burch. 2019. Complexity-weighted loss and diverse reranking for sentence simplification. 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 3137–3147.

Mateusz Krubiński, Erfan Ghadery, Marie-Francine Moens, and Pavel Pecina. 2021. Just ask! evaluating machine translation by asking and answering questions. In *Proceedings of the Sixth Conference on Machine Translation*, pages 495–506, Online. Association for Computational Linguistics.

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. 2022. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. In *The Eleventh International Conference on Learning Representations*.

Dhruv Kumar, Lili Mou, Lukasz Golab, and Olga Vechtomova. 2020. Iterative edit-based unsupervised sentence simplification. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7918–7928, Online. Association for Computational Linguistics.

Gaurav Kumar, George Foster, Colin Cherry, and Maxim Krikun. 2019. Reinforcement learning based curriculum optimization for neural machine translation. 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 2054–2061, Minneapolis, Minnesota. Association for Computational Linguistics.

Sachin Kumar, Eric Malmi, Aliaksei Severyn, and Yulia Tsvetkov. 2021. Controlled text generation as continuous optimization with multiple constraints. *Advances in Neural Information Processing Systems*, 34:14542–14554.

Philippe Laban, Tobias Schnabel, Paul Bennett, and Marti A. Hearst. 2021. Keep it simple: Unsupervised simplification of multi-paragraph text. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6365–6378, Online. Association for Computational Linguistics.

Guillaume Lample, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2018. Unsupervised machine translation using monolingual corpora only. In *International Conference on Learning Representations*.

Rémi Leblond, Jean-Baptiste Alayrac, Anton Osokin, and Simon Lacoste-Julien. 2017. Searnn: Training rnns with global-local losses. *arXiv preprint arXiv:1706.04499*.

Rémi Leblond, Jean-Baptiste Alayrac, Anton Osokin, and Simon Lacoste-Julien. 2018. Searnn: Training rnns with global-local losses. In *International Conference on Learning Representations*.

Jason Lee, Elman Mansimov, and Kyunghyun Cho. 2018. Deterministic non-autoregressive neural sequence modeling by iterative refinement. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1173–1182.

Oliver Lemon, Srini Janarthanam, and Verena Rieser. 2010. Generation under uncertainty. In *Proceedings of the 6th International Natural Language Generation Conference*.

Gondy Leroy, David Kauchak, Diane Haeger, and Douglas Spegman. 2022. Evaluation of an online text simplification editor using manual and automated metrics for perceived and actual text difficulty. *JAMIA Open*, 5(2):ooac044.

Vladimir I Levenshtein et al. 1966. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, volume 10, pages 707–710. Soviet Union.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Jialu Li, Esin Durmus, and Claire Cardie. 2020. Exploring the role of argument structure in online debate persuasion. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8905–8912, Online. Association for Computational Linguistics.

Xiang Lisa Li, John Thickstun, Ishaan Gulrajani, Percy Liang, and Tatsunori B Hashimoto. 2022. Diffusion-lm improves controllable text generation. *arXiv preprint arXiv:2205.14217*.

Chin-Yew Lin and Franz Josef Och. 2004. ORANGE: a method for evaluating automatic evaluation metrics for machine translation. In *COLING 2004: Proceedings of the 20th International Conference on Computational Linguistics*, pages 501–507, Geneva, Switzerland. COLING.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. Teaching models to express their uncertainty in words. *Transactions on Machine Learning Research*.

Xi Victoria Lin, Todor Mihaylov, Mikel Artetxe, Tianlu Wang, Shuohui Chen, Daniel Simig, Myle Ott, Naman Goyal, Shruti Bhosale, Jingfei Du, et al. 2021. Few-shot learning with multilingual language models. *arXiv preprint arXiv:2112.10668*.

Jinglin Liu, Yi Ren, Xu Tan, Chen Zhang, Tao Qin, Zhou Zhao, and Tie-Yan Liu. 2021a. Task-level curriculum learning for non-autoregressive neural machine translation. In *Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence*, pages 3861–3867.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2021b. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *arXiv preprint arXiv:2107.13586*.

Chi-kiu Lo. 2019. YiSi - a unified semantic MT quality evaluation and estimation metric for languages with different levels of available resources. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 507–513, Florence, Italy. Association for Computational Linguistics.

Mengsay Loem, Masahiro Kaneko, Sho Takase, and Naoaki Okazaki. 2023. Exploring effectiveness of gpt-3 in grammatical error correction: A study on performance and controllability in prompt-based methods. *arXiv preprint arXiv:2305.18156*.

Michael H. Long and Steven J. Ross. 1993. Modifications that preserve language and content.

Qingsong Ma, Johnny Wei, Ondřej Bojar, and Yvette Graham. 2019. Results of the WMT19 metrics shared task: Segment-level and strong MT systems pose big challenges. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 62–90, Florence, Italy. Association for Computational Linguistics.

Mounica Maddela, Fernando Alva-Manchego, and Wei Xu. 2020. Controllable text simplification with explicit paraphrasing. *arXiv preprint arXiv:2010.11004*.

Mounica Maddela, Fernando Alva-Manchego, and Wei Xu. 2021. Controllable text simplification with explicit paraphrasing. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3536–3553, Online. Association for Computational Linguistics.

Mounica Maddela, Yao Dou, David Heineman, and Wei Xu. 2022. Lens: A learnable evaluation metric for text simplification. *arXiv preprint arXiv:2212.09739*.

Jonathan Mallinson and Mirella Lapata. 2019. Controllable sentence simplification: Employing syntactic and lexical constraints. *arXiv preprint arXiv:1910.04387*.

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 5057–5068.

Louis Martin, Éric de la Clergerie, Benoît Sagot, and Antoine Bordes. 2020a. Controllable sentence simplification. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 4689–4698, Marseille, France. European Language Resources Association.

Louis Martin, Éric Villemonte de la Clergerie, Benoît Sagot, and Antoine Bordes. 2020b. Controllable sentence simplification. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 4689–4698.

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.

Louis Martin, Benoît Sagot, Éric de la Clergerie, and Antoine Bordes. 2019. Controllable sentence simplification. *arXiv preprint arXiv:1910.02677*.

Stephen Mayhew, Klinton Bicknell, Chris Brust, Bill McDowell, Will Monroe, and Burr Settles. 2020. Simultaneous translation and paraphrase for language education. In *Proceedings of the Fourth Workshop on Neural Generation and Translation*, pages 232–243, Online. Association for Computational Linguistics.

Mary L McHugh. 2012. Interrater reliability: the kappa statistic. *Biochemia medica*, 22(3):276–282.

Larry R Medsker and LC Jain. 2001. Recurrent neural networks. *Design and Applications*, 5:64–67.

Sneha Mehta, Bahareh Azarnoush, Boris Chen, Avneesh Saluja, Vinith Misra, Ballav Bihani, and Ritwik Kumar. 2020. Simplify-then-translate: Automatic preprocessing for black-box translation. In *AAAI*.

Paul Michel and Graham Neubig. 2018. Extreme adaptation for personalized neural machine translation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 312–318, Melbourne, Australia. Association for Computational Linguistics.

John Milton. 2009. Translation studies and adaptation studies. *Translation research projects*, 2:51–58.

Shachar Mirkin and Jean-Luc Meunier. 2015. Personalized Machine Translation: Predicting Translational Preferences. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2019–2025, Lisbon, Portugal. Association for Computational Linguistics.

Makoto Miwa, Rune Sætre, Yusuke Miyao, and Jun'ichi Tsujii. 2010. Entity-focused sentence simplification for relation extraction. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 788–796, Beijing, China. Coling 2010 Organizing Committee.

Rei Miyata and Midori Tatsumi. 2019. Evaluating the suitability of human-oriented text simplification for machine translation. In *Proceedings of the 33rd Pacific Asia Conference on Language, Information and Computation*.

Dragos Stefan Munteanu and Daniel Marcu. 2005. Improving machine translation performance by exploiting non-parallel corpora. *Computational Linguistics*, 31(4):477–504.

Maria Nădejde, Anna Currey, Benjamin Hsu, Xing Niu, Marcello Federico, and Georgiana Dinu. 2022. Cocoa-mt: A dataset and benchmark for contrastive controlled mt with application to formality. *arXiv preprint arXiv:2205.04022*.

Graham Neubig. 2017. Neural machine translation and sequence-to-sequence models: A tutorial. *arXiv preprint arXiv:1703.01619*.

Graham Neubig, Makoto Morishita, and Satoshi Nakamura. 2015. Neural reranking improves subjective quality of machine translation: Naist at wat2015. In *Proceedings of the 2nd Workshop on Asian Translation (WAT2015)*, pages 35–41.

Daiki Nishihara, Tomoyuki Kajiwara, and Yuki Arase. 2019. Controllable text simplification with lexical constraint loss. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 260–266, Florence, Italy. Association for Computational Linguistics.

Sergiu Nisioi, Sanja Štajner, Simone Paolo Ponzetto, and Liviu P. Dinu. 2017. Exploring neural text simplification models. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 85–91, Vancouver, Canada. Association for Computational Linguistics.

Xing Niu, Sudha Rao, and Marine Carpuat. 2018a. Multi-Task Neural Models for Translating Between Styles Within and Across Languages. In *27th International Conference on Computational Linguistics (COLING 2018)*.

Xing Niu, Sudha Rao, and Marine Carpuat. 2018b. Multi-task neural models for translating between styles within and across languages. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1008–1021, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

SUN-YOUNG OH. 2001. Two types of input modification and efl reading comprehension: Simplification versus elaboration. *TESOL quarterly*, 35(1):69–96.

Myle Ott, Michael Auli, David Grangier, and Marc'Aurelio Ranzato. 2018. Analyzing uncertainty in neural machine translation. In *International Conference on Machine Learning*, pages 3956–3965. PMLR.

Gustavo Paetzold, Fernando Alva-Manchego, and Lucia Specia. 2017. Massalign: Alignment and annotation of comparable documents. *Proceedings of the IJCNLP 2017, System Demonstrations*, pages 1–4.

Gustavo Henrique Paetzold and Lucia Specia. 2016. Semeval 2016 task 11: Complex word identification. *Proceedings of SemEval*, pages 560–569.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002a. 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.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002b. 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.

Amandalynne Paullada, Inioluwa Deborah Raji, Emily M Bender, Emily Denton, and Alex Hanna. 2021. Data and its (dis) contents: A survey of dataset development and use in machine learning research. *Patterns*, 2(11).

Sarah E Petersen and Mari Ostendorf. 2007. Text simplification for language learners: a corpus analysis. In *Workshop on Speech and Language Technology in Education*.

Emmanouil Antonios Platanios, Otilia Stretcu, Graham Neubig, Barnabás Póczos, and Tom M Mitchell. 2019. Competence-based curriculum learning for neural machine translation. In *NAACL-HLT (1)*.

Maja Popović. 2015. chrF: character n-gram F-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395, Lisbon, Portugal. Association for Computational Linguistics.

Matt Post and David Vilar. 2018. Fast lexically constrained decoding with dynamic beam allocation for neural machine translation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1314–1324.

Shrimai Prabhumoye, Alan W Black, and Ruslan Salakhutdinov. 2020. Exploring controllable text generation techniques. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 1–14, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Lihua Qian, Hao Zhou, Yu Bao, Mingxuan Wang, Lin Qiu, Weinan Zhang, Yong Yu, and Lei Li. 2020. Glancing transformer for non-autoregressive neural machine translation. *arXiv e-prints*, pages arXiv–2008.

Yu Qiao, Xiaofei Li, Daniel Wiechmann, and Elma Kerz. 2022. (psycho-)linguistic features meet transformer models for improved explainable and controllable text simplification. In *Proceedings of the Workshop on Text Simplification, Accessibility, and Readability (TSAR-2022)*, pages 125–146, Abu Dhabi, United Arab Emirates (Virtual). Association for Computational Linguistics.

Ying Qin and Lucia Specia. 2015. Truly exploring multiple references for machine translation evaluation. In *Proceedings of the 18th Annual Conference of the European Association for Machine Translation*, Antalya, Turkey. European Association for Machine Translation.

Ella Rabinovich, Shachar Mirkin, Raj Nath Patel, Lucia Specia, and Shuly Wintner. 2016. Personalized Machine Translation: Preserving Original Author Traits. *arXiv:1610.05461 [cs]*.

Sudha Rao and Joel Tetreault. 2018. Dear sir or madam, may I introduce the GYAFC dataset: Corpus, benchmarks and metrics for formality style transfer. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 129–140, New Orleans, Louisiana. Association for Computational Linguistics.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. COMET: A neural framework for MT evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2685–2702, Online. Association for Computational Linguistics.

Philip Resnik. 1998. Parallel strands: A preliminary investigation into mining the web for bilingual text. In *Conference of the Association for Machine Translation in the Americas*, pages 72–82. Springer.

Irina Rets and Jekaterina Rogaten. 2021. To simplify or not? facilitating english l2 users' comprehension and processing of open educational resources in english using text simplification. *Journal of Computer Assisted Learning*, 37(3):705–717.

Anna Rogers. 2021. Changing the world by changing the data. *arXiv preprint arXiv:2105.13947*.

Stephane Ross and Drew Bagnell. 2010. Efficient reductions for imitation learning. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pages 661–668, Chia Laguna Resort, Sardinia, Italy. PMLR.

Horacio Saggion. 2017. Automatic text simplification. In *Synthesis Lectures on Human Language Technologies*.

Chitwan Saharia, William Chan, Saurabh Saxena, and Mohammad Norouzi. 2020. Non-autoregressive machine translation with latent alignments. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1098–1108, Online. Association for Computational Linguistics.

Marina Sanchez-Torron and Philipp Koehn. 2016. Machine translation quality and post-editor productivity. In *Conferences of the Association for Machine Translation in the Americas: MT Researchers' Track*, pages 16–26, Austin, TX, USA. The Association for Machine Translation in the Americas.

Carolina Scarton and Lucia Specia. 2016. A reading comprehension corpus for machine translation evaluation. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 3652–3658, Portorož, Slovenia. European Language Resources Association (ELRA).

Carolina Scarton and Lucia Specia. 2018a. Learning Simplifications for Specific Target Audiences. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 712–718.

Carolina Scarton and Lucia Specia. 2018b. 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.

Yves Scherrer. 2020. Tapaco: A corpus of sentential paraphrases for 73 languages. In *Proceedings of The 12th Language Resources and Evaluation Conference*. European Language Resources Association (ELRA).

Andrea Schioppa, David Vilar, Artem Sokolov, and Katja Filippova. 2021. Controlling machine translation for multiple attributes with additive interventions. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6676–6696, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Jordan Schmidek and Denilson Barbosa. 2014. Improving open relation extraction via sentence re-structuring. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 3720–3723, Reykjavik, Iceland. European Language Resources Association (ELRA).

Elliot Schumacher, Maxine Eskenazi, Gwen Frishkoff, and Kevyn Collins-Thompson. 2016. Predicting the relative difficulty of single sentences with and without surrounding context. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1871–1881, Austin, Texas. Association for Computational Linguistics.

Max Schwarzer and David Kauchak. 2018. Human evaluation for text simplification: The simplicity-adequacy tradeoff. In *SoCal NLP Symposium*.

Thibault Sellam, Dipanjan Das, and Ankur Parikh. 2020. BLEURT: Learning robust metrics for text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7881–7892, Online. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Controlling Politeness in Neural Machine Translation via Side Constraints. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 35–40. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. Neural Machine Translation of Rare Words with Subword Units. *Proceedings of the Meeting of the Association for Computational Linguistics (ACL)*.

RJ Senter and Edgar A Smith. 1967. Automated readability index. Technical report, CINCINNATI UNIV OH.

Matthew Shardlow. 2014. A survey of automated text simplification. *International Journal of Advanced Computer Science and Applications*, 4(1):58–70.

Kim Cheng Sheang, Daniel Ferrés, and Horacio Saggion. 2022. Controllable lexical simplification for English. In *Proceedings of the Workshop on Text Simplification, Accessibility, and Readability (TSAR-2022)*, pages 199–206, Abu Dhabi, United Arab Emirates (Virtual). Association for Computational Linguistics.

Kim Cheng Sheang and Horacio Saggion. 2021. Controllable sentence simplification with a unified text-to-text transfer transformer. In *Proceedings of the 14th International Conference on Natural Language Generation*, pages 341–352, Aberdeen, Scotland, UK. Association for Computational Linguistics.

Shiqi Shen, Yong Cheng, Zhongjun He, Wei He, Hua Wu, Maosong Sun, and Yang Liu. 2016. Minimum risk training for neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1683–1692, Berlin, Germany. Association for Computational Linguistics.

Raphael Shu, Hideki Nakayama, and Kyunghyun Cho. 2019. Generating diverse translations with sentence codes. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1823–1827.

Advaith Siddharthan. 2014. A survey of research on text simplification. *ITL - International Journal of Applied Linguistics*, 165(2):259–298.

Advaith Siddharthan and Napoleon Katsos. 2010. Reformulating discourse connectives for non-expert readers. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 1002–1010. Association for Computational Linguistics.

Lucia Specia. 2010. Translating from complex to simplified sentences. In *International Conference on Computational Processing of the Portuguese Language*, pages 30–39. Springer.

Lucia Specia and Kashif Shah. 2018. *Machine Translation Quality Estimation: Applications and Future Perspectives*. Springer International Publishing, Cham.

Sanja Stajner. 2021. Automatic text simplification for social good: Progress and challenges. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2637–2652, Online. Association for Computational Linguistics.

Sanja Štajner, Marc Franco-Salvador, Paolo Rosso, and Simone Paolo Ponzetto. 2018. Cats: A tool for customized alignment of text simplification corpora. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC-2018)*.

Sanja Štajner and Maja Popovic. 2016. Can text simplification help machine translation? In *Proceedings of the 19th Annual Conference of the European Association for Machine Translation*, pages 230–242.

Sanja Štajner and Maja Popović. 2018. Improving machine translation of English relative clauses with automatic text simplification. In *Proceedings of the 1st Workshop on Automatic Text Adaptation (ATA)*, pages 39–48, Tilburg, the Netherlands. Association for Computational Linguistics.

Mitchell Stern, William Chan, Jamie Kiros, and Jakob Uszkoreit. 2019. Insertion transformer: Flexible sequence generation via insertion operations. In *International Conference on Machine Learning*, pages 5976–5985. PMLR.

Saku Sugawara, Kentaro Inui, Satoshi Sekine, and Akiko Aizawa. 2018. What makes reading comprehension questions easier? In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4208–4219, Brussels, Belgium. Association for Computational Linguistics.

Raymond Hendy Susanto, Shamil Chollampatt, and Liling Tan. 2020. Lexically constrained neural machine translation with Levenshtein transformer. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3536–3543, Online. Association for Computational Linguistics.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. *Advances in neural information processing systems*, 27.

Irina Temnikova and Galina Maneva. 2013. The C-score – proposing a reading comprehension metrics as a common evaluation measure for text simplification. In *Proceedings of the Second Workshop on Predicting and Improving Text Readability for Target Reader Populations*, pages 20–29, Sofia, Bulgaria. Association for Computational Linguistics.

Irina Temnikova, Constantin Orasan, and Ruslan Mitkov. 2012. CLCM - a linguistic resource for effective simplification of instructions in the crisis management domain and its evaluations. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 3007–3014, Istanbul, Turkey. European Language Resources Association (ELRA).

Jörg Tiedemann. 2009. News from OPUS-A collection of multilingual parallel corpora with tools and interfaces. In *Recent Advances in Natural Language Processing*, volume 5, pages 237–248.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in opus. In *Lrec*, volume 2012, pages 2214–2218.

Sara Tonelli, Alessio Palmero Aprosio, and Francesca Saltori. 2016. Simpitiki: a simplification corpus for italian. In *CLiC-it/EVALITA*.

Adel I Tweissi. 1998. The effects of the amount and type of simplification on foreign language reading comprehension.

Sowmya Vajjala and Ivana Lučić. 2018. OneStopEnglish corpus: A new corpus for automatic readability assessment and text simplification. In *Proceedings of the Thirteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 297–304, New Orleans, Louisiana. Association for Computational Linguistics.

Sowmya Vajjala and Ivana Lucic. 2019. On understanding the relation between expert annotations of text readability and target reader comprehension. In *Proceedings of the Fourteenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 349–359, Florence, Italy. Association for Computational Linguistics.

Hoang Van, Zheng Tang, and Mihai Surdeanu. 2021. How may I help you? using neural text simplification to improve downstream NLP tasks. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 4074–4080, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

David Vickrey and Daphne Koller. 2008. Sentence simplification for semantic role labeling. In *Proceedings of ACL-08: HLT*, pages 344–352, Columbus, Ohio. Association for Computational Linguistics.

Yogarshi Vyas, Xing Niu, and Marine Carpuat. 2018. Identifying semantic divergences in parallel text without annotations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language*

Technologies, Volume 1 (Long Papers), pages 1503–1515. Association for Computational Linguistics.

David Wan, Chris Kedzie, Faisal Ladhak, Marine Carpuat, and Kathleen McKeown. 2020. Incorporating terminology constraints in automatic post-editing. In *Proceedings of the Fifth Conference on Machine Translation*, pages 1193–1204.

Alex Wang, Kyunghyun Cho, and Mike Lewis. 2020. Asking and answering questions to evaluate the factual consistency of summaries. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5008–5020, Online. Association for Computational Linguistics.

Tong Wang, Ping Chen, John Rochford, and Jipeng Qiang. 2016. Text simplification using neural machine translation. In *Thirtieth AAAI Conference on Artificial Intelligence*.

Willian Massami Watanabe, Arnaldo Candido Junior, Vinícius Rodriguez Uzêda, Renata Pontin de Mattos Fortes, Thiago Alexandre Salgueiro Pardo, and Sandra Maria Aluísio. 2009. Facilita: reading assistance for low-literacy readers. In *Proceedings of the 27th ACM international conference on Design of communication*, pages 29–36.

Warren Weaver. 1952. Translation. In *Proceedings of the Conference on Mechanical Translation*, Massachusetts Institute of Technology.

Laura Weidinger, Jonathan Uesato, Maribeth Rauh, Conor Griffin, Po-Sen Huang, John Mellor, Amelia Glaese, Myra Cheng, Borja Balle, Atoosa Kasirzadeh, et al. 2022. Taxonomy of risks posed by language models. In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*, pages 214–229.

Sean Welleck, Ilia Kulikov, Stephen Roller, Emily Dinan, Kyunghyun Cho, and Jason Weston. 2019. Neural text generation with unlikelihood training. In *International Conference on Learning Representations*.

John Wieting, Taylor Berg-Kirkpatrick, Kevin Gimpel, and Graham Neubig. 2019. Beyond BLEU:training neural machine translation with semantic similarity. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4344–4355, Florence, Italy. Association for Computational Linguistics.

Sander Wubben, Antal van den Bosch, and Emiel Krahmer. 2012. Sentence simplification by monolingual machine translation. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1015–1024, Jeju Island, Korea. Association for Computational Linguistics.

Yijun Xiao and William Yang Wang. 2021. On hallucination and predictive uncertainty in conditional language generation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2734–2744, Online. Association for Computational Linguistics.

Wei Xu, Chris Callison-Burch, and Courtney Napoles. 2015. Problems in current text simplification research: New data can help. *Transactions of the Association for Computational Linguistics*, 3:283–297.

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.

Weijia Xu, Sweta Agrawal, Eleftheria Briakou, Marianna Martindale, and Marine Carpuat. 2023. Understanding and detecting hallucinations in neural machine translation via model introspection. *Transactions of the Association for Computational Linguistics*, 11.

Weijia Xu and Marine Carpuat. 2021. Editor: An edit-based transformer with repositioning for neural machine translation with soft lexical constraints. *Transactions of the Association for Computational Linguistics*, 9:311–328.

Daiki Yanamoto, Tomoki Ikawa, Tomoyuki Kajiwara, Takashi Ninomiya, Satoru Uchida, and Yuki Arase. 2022. Controllable text simplification with deep reinforcement learning. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 398–404, Online only. Association for Computational Linguistics.

Ya-Han Yang, Hsi-Chin Chu, and Wen-Ta Tseng. 2021. Text difficulty in extensive reading: Reading comprehension and reading motivation. *Reading in a Foreign Language*, 33(1):78–102.

Elizaveta Yankovskaya, Andre Tättar, and Mark Fishel. 2019. Quality estimation and translation metrics via pre-trained word and sentence embeddings. In *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, pages 101–105, Florence, Italy. Association for Computational Linguistics.

Ziyu Yao, Frank F. Xu, Pengcheng Yin, Huan Sun, and Graham Neubig. 2021. Learning structural edits via incremental tree transformations. In *International Conference on Learning Representations*.

Dolly N Young. 1999. Linguistic simplification of sl reading material: Effective instructional practice? *The Modern Language Journal*, 83(3):350–366.

Tatsuya Zetsu, Tomoyuki Kajiwara, and Yuki Arase. 2022. Lexically constrained decoding with edit operation prediction for controllable text simplification. In *Proceedings of the Workshop on Text Simplification, Accessibility, and Readability (TSAR-2022)*, pages 147–153, Abu Dhabi, United Arab Emirates (Virtual). Association for Computational Linguistics.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

Xingxing Zhang and Mirella Lapata. 2017. Sentence simplification with deep reinforcement learning. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 584–594, Copenhagen, Denmark. Association for Computational Linguistics.

Xuan Zhang, Gaurav Kumar, Huda Khayrallah, Kenton Murray, Jeremy Gwinnup, Marianna J Martindale, Paul McNamee, Kevin Duh, and Marine Carpuat. 2018a. An empirical exploration of curriculum learning for neural machine translation.

Yizhe Zhang, Michel Galley, Jianfeng Gao, Zhe Gan, Xiujun Li, Chris Brockett, and Bill Dolan. 2018b. Generating informative and diverse conversational responses via adversarial information maximization. In *Advances in Neural Information Processing Systems*, pages 1810–1820.

Wei Zhao, Goran Glavaš, Maxime Peyrard, Yang Gao, Robert West, and Steffen Eger. 2020. On the limitations of cross-lingual encoders as exposed by reference-free machine translation evaluation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1656–1671, Online. Association for Computational Linguistics.

Yikai Zhou, Baosong Yang, Derek F. Wong, Yu Wan, and Lidia S. Chao. 2020. Uncertainty-aware curriculum learning for neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6934–6944, Online. Association for Computational Linguistics.

Johannes C Ziegler, Caroline Castel, Catherine Pech-Georgel, Florence George, F-Xavier Alario, and Conrad Perry. 2008. Developmental dyslexia and the dual route model of reading: Simulating individual differences and subtypes. *Cognition*, 107(1):151–178.

Vilém Zouhar, Martin Popel, Ondřej Bojar, and Aleš Tamchyna. 2021. Neural machine translation quality and post-editing performance. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10204–10214, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.