Medical Text Simplification: Bridging the Gap between Medical Research and Public Understanding

by

Atharva Phatak

A thesis submitted in partial fulfillment of the

requirements for the degree of

Master of Science

in

Computer Science

in the

Faculty of Science and Environmental Studies

of

Lakehead University, Thunder Bay

Committee in charge:

Dr. Vijay Mago (Principal Supervisor)

Dr. Ameeta Agrawal (External Examiner)
Dr. Garima Bajwa (Internal Examiner)

Winter 2023

The thesis of Atharva Phatak, titled Medical Text Simplification: Bridging the Gap between Medical Research and Public Understanding, is approved:

Chair _____ Date _____

_____ Date _____

_____ Date _____

Lakehead University, Thunder Bay

DECLARATION

I certify that,

- The work contained in this thesis is original and has been done by myself and under the general supervision of my supervisor(s).
- The work reported herein has not been submitted to any other Institute for any degree or diploma.
- Whenever I have used materials (concepts, ideas, text, expressions, data, graphs, diagrams, theoretical analysis, results, etc.) from other sources, I have given due credit by citing them in the text of the thesis and giving their details in the references. Elaborate sentences used verbatim from published work have been clearly identified and quoted.
- I also affirm that no part of this thesis can be considered plagiarism to the best of my knowledge and understanding and take complete responsibility if any complaint arises.
- I am fully aware that my thesis supervisor(s) are not in a position to check for any possible instance of plagiarism within this submitted work.

Medical Text Simplification: Bridging the Gap between Medical Research and Public Understanding

Atharva Phatak

Abstract

MEDICAL TEXT SIMPLIFICATION: BRIDGING THE GAP BETWEEN MEDICAL RESEARCH AND PUBLIC UNDERSTANDING

Text Simplification is a subdomain of Natural Language Processing that focuses on applying computational techniques to modify the content and structure of the text to make it interpretable while retaining the main idea. The advancements in text simplification research have provided valuable benefits to a wide range of readers, including those with learning disabilities and non-native speakers. Moreover, even regular readers who are not experts in fields such as medicine or finance have found text simplification techniques to be useful in accessing scientific literature and research. This thesis aims to create a text simplification approach that can effectively simplify complex biomedical literature. Chapter 2 provides an insightful overview of the datasets, methods, and evaluation techniques used in text simplification. Chapter 3 conducts an extensive bibliometric analysis of literature in the field of text simplification to understand research trends, find important research and application topics of text simplification research, and understand shortcomings in the field. Based on the findings in Chapter 3, we found that the advancements in text simplification research can have a positive impact on the medical domain. The research in the field of medicine is constantly developing and contains important information about drugs and treatments for various life threatening diseases. Although this information is accessible to the public, it is very complex in nature, thus making it difficult to understand. To address this problem, chapter 4 proposes an Automatic Text Simplification approach called "TESLEA", which is capable of simplifying text related to the medical domain. The proposed approach employs

a transformer-based model and leverages reinforcement learning to train the model in optimizing rewards that are tailored to text simplification. The proposed method outperformed previous baselines on Flesch-Kincaid scores (11.84) and achieved comparable performance with other baselines when measured using ROUGE-1 (0.39), ROUGE-2 (0.11), and SARI scores (0.40). The analysis of human annotated data revealed a percentage agreement of over 70% among human annotators when evaluated factors such as fluency, coherence, and adequacy. While having proposed an approach for simplifying medical text, this research also identifies potential avenues for future investigation, specifically the development of multilingual text simplification systems catering to diverse domains.

Contents

List of Figures

List of Tables

Acknowledgments

First and foremost, I express my sincere gratitude to my supervisor Dr. Vijay Mago for his academic, financial, and moral support, without which this research would not have been possible. I also extend my gratitude to Dr. David Savage (NOSM University) and all of my colleagues at DaTALab in the CASES building at Lakehead University. I am grateful for all the resources made available to me at DaTALab, and the NSERC Discovery Grant held by my supervisor, for supporting me throughout my degree. Finally, I would like to express my heartfelt gratitude to my parents and friends for always believing in me and supporting my aspirations.

Chapter 1

Introduction

In recent years, Natural Language Processing (NLP) has experienced noteworthy advancements, primarily attributable to the introduction of transformer-based models. These models have demonstrated remarkable success in accomplishing state-of-the-art performance on numerous Natural Language Generation (NLG) tasks like Text summarization and Question Answering. One of the most researched tasks in NLP is Text Simplification (TS), which aims to employ various computational techniques to transform the contents of complex text into a simplified version, thereby facilitating ease of comprehension and ensuring that the core idea of the original text is retained. The field of text simplification has made significant progress in recent years, bringing about a multitude of benefits for a diverse range of readers. These advancements have proven particularly valuable for individuals who are non-native speakers of a language or dialect, as well as those who have learning disabilities. However, even regular readers who lack expertise in fields such as finance or medicine have also benefited greatly from the application of text simplification techniques.

The objective of this thesis is as follows

- To conduct a comprehensive analysis of existing research on Text Simplification (TS)

using Bibliometric analysis techniques to understand the contribution of authors and organizations to specific topics, influential studies in the field and the connections between them, and the trends of a particular research field.

- To develop an approach that leverages state-of-the-art Language Models (LMs) to develop an automatic TS solution capable of simplifying complex medical vocabulary found in research articles related to healthcare.

In Chapter 2 of the thesis, readers will gain valuable knowledge on key components that are essential for understanding the subsequent chapters. This chapter delves into three main areas: datasets, evaluation metrics, and language models. Specifically, it provides insights on how standard datasets in the field of TS are constructed and highlights the properties and training approaches of various state-of-the-art language models. Additionally, readers will be introduced to the most commonly used evaluation metrics in the field of TS. Overall, Chapter 2 is a crucial resource for readers seeking a deeper understanding of the foundations of TS research.

The Chapter 3 of this thesis, we conduct an extensive analysis of Text Simplification literature via the help of Bibliometric analysis. This analysis is designed to answer six research questions that aim to analyze the progress of research in the field of TS, important persona in the field of TS, the collaboration between the authors and how it has evolved with time and finally important research and application topics in the field of TS. The dataset for this analysis was collected from Google Scholar (GS) with the help of "Publish Perish" software and Scholarly API. Articles having "Text Simplification" in their titles were selected from years 2001-2022, resulting in a dataset of 656 articles and additional metadata associated with articles was also collected. Collaboration and Temporal Analysis was applied to understand collaborations and the evolution of collaborations in the field. Topic Modeling

and Word Frequency analysis was done to uncover important research and application topics. The findings from Chapter 3 revealed that text simplification techniques are being utilized in the field of bio-medicine. More specifically, the research articles within the medical domain are available to the public. However, even though these articles are accessible to everyone, they are often difficult to understand for a broader audience due to the complex medical terminology used in them. As a result, simplifying these complex abstracts is crucial to ensure that medical research is comprehensible to the general public.

To tackle the problem of medical text simplification, Chapter 4 proposes a novel deep learning based text simplification approach that converts complex medical text to a simpler version while maintaining the quality of the generated text. This approach uses a transformer based language model called BART as a text generator and trains the model using a combination of standard finetuning (domain adaptation) and Reinforcement Learning (RL) to optimize TS specific rewards that capture the properties of simplicity and relevance. In addition, the training process of both standard fine-tuning and reinforcement learning provides a universal framework. This is because the framework is not specific to any particular model, meaning that any autoregressive language model can be substituted for BART. Additionally, the framework can be modified to work with other datasets by making minor changes to the reward functions. Based on comprehensive analysis conducted to evaluate the models performance, it was observed that the proposed method outperformed previous baselines on Flesch Kincaid Scores (11.84) and achieved comparable performance to other baselines when measured using ROUGE-1 (0.39), ROUGE-2 (0.11) and SARI scores (0.40). According to the results of manual evaluation, when factors such as fluency, coherence, and adequacy were taken into account, the percent agreement between human annotators was found to be more than 70%. This suggests that the human annotators generally agreed with each other when assessing these factors in the content being evaluated.

The Chapter 5 of this thesis concludes by highlighting the shortcomings of text simplification models and the persisting challenges that confront the domain of text simplification. The primary contribution of this thesis

- An extensive bibliometric analysis of Text Simplification literature was conducted along with development of a website [1] that highlights the analysis done to answer six research questions.
- A universal approach that combines finetuning and reinforcement learning based training to train a cutting edge language model (BART) to optimize TS specific rewards so that the model is capable of simplifying complex medical text data.

The research conducted during this work is open-sourced and readily available in a Github Repository [2].

[1] https://bblts.datalab.science/
[2] https://github.com/Atharva-Phatak/TESLEA

Chapter 2

Literature Review: Text Simplification

2.1 Introduction

The field of Natural Language Processing (NLP) has seen a significant rise in recent years due to the progress in the field of Deep Learning (DL). In recent years sub-fields of NLP like text summarization, text simplification, semantic similarity [51], etc, have progressed significantly in research leading to the development of various applications which leverage NLP at scale. Text Simplification aims to convert a difficult-to-understand text in such a way that it becomes more readable, easy to understand, and retains the main idea of the text. In the early days, TS research focused on Lexical Simplification (LS) [10, 48]. A lexical simplification system typically involves identifying and replacing complex words with their simpler alternatives [71]. Recent research defines TS as a sequence-to-sequence (Seq2Seq) task and has tackled it by leveraging model architectures from other text generation domains like machine translation or text summarization. Moreover, TS is a field with diverse

applications catering to audiences of various domains. The most prominent target audiences for TS are non-native readers/foreign language learners for whom various applications for simplifying text have been developed [72, 33]. This is all possible because of the development of Automatic Text Simplification approaches (ATS), which leverage modern seq2seq model architectures. The goal of this chapter is to provide background about text simplification datasets, evaluation metrics and current models.

2.2 Text Simplification Datasets

The latest techniques in TS are heavily data-driven, taking advantage of recent approaches in NLG. Most of the techniques for ATS require a parallel dataset consisting of complex text and simple text. Most of the datasets used in the field of TS are sentence-level datasets (i.e., each pair in the dataset consists of a complex sentence and a simple sentence), whereas the recent research in the field of TS has been focusing on the construction of paragraphs or document level datasets, multilingual datasets, domain-specific datasets and datasets designed for evaluation purposes. This section will provide brief introductions about commonly used datasets in the field of TS.

Sentence Level Datasets

Most of the sentence-level datasets have been extracted from majorly two data sources 1) Simple English Wikipedia (SEW) and 2) Newsela Corpus.

- **EW-SEW** [15]: This dataset was created by aligning sentences from the paired articles extracted from English Wikipedia(EW) and Simple English Wikipedia(SEW). The first step was pairing articles from EW and SEW which was done with the help of titles. A

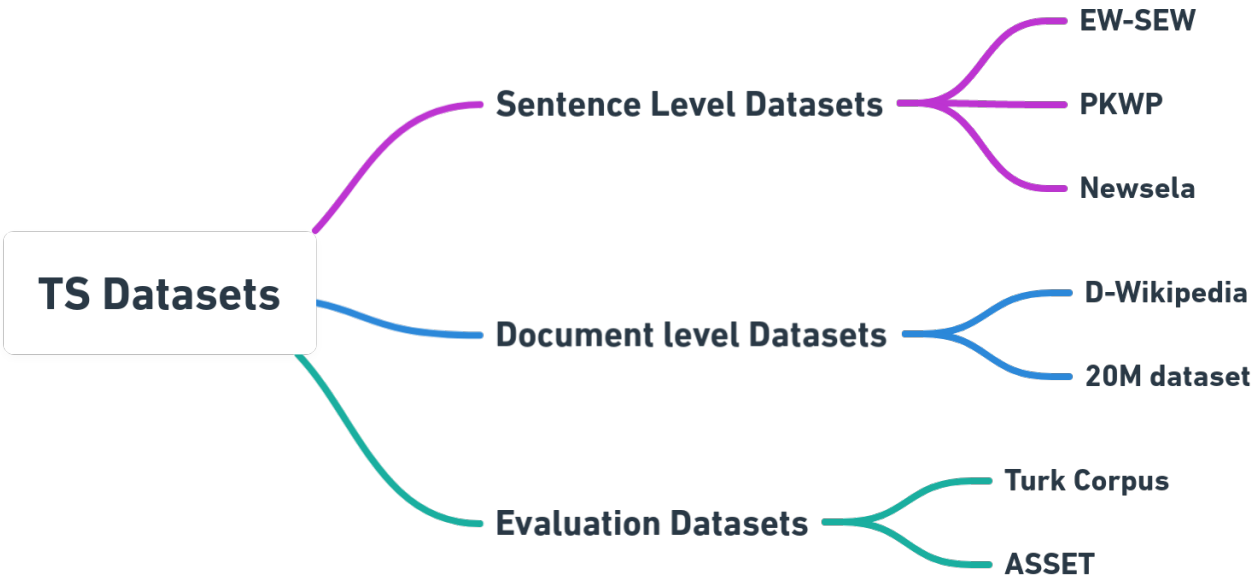

Figure 2.1: Standard Datasets used in the field of TS.

cleaning step was done to remove noisy articles which involved removing articles which contained only a single line, articles marked as disambiguation or articles which were flagged as stub. After the filtering process, 10588 articles pairs remained. These article pairs were then used to extract paired paragraphs based on the formatting information. Every normal paragraph was aligned to its corresponding simple paragraph where the TF-IDF cosine-similarity was above the threshold of 0.5. Finally to extract aligned sentences from the paired simplex-complex paragraphs a dynamic programming-based approach proposed by Barzilay and Elhadad was used which resulted in extraction of 137K aligned sentence pairs. The aligned sentence pairs cover main simplification operations of rewording, reordering, insertion and deletion. This dataset is one of the most widely used datasets in TS tasks. [15]

- **Parallel Wikipedia Simplification (PKWP)** [86]: The PKWP dataset was also extracted from EW and SEW but the methodology used to extract article pairs and align sentences was different than Coster and Kauchak. The EW and SEW article pairs were aligned by using "language link" available in wikimedia dump files. A text extraction procedure called JWPL [81] was used on these articles to extract plain text from these article pairs. Further, preprocessing steps like sentence boundary detection, tokenization and lemmatization were also used. Finally, to align sentence pairs from these articles three similarity measures were used namely, sentence level TF-IDF [44], Word Overlap [5] and Word based edit distance. To evaluate the effectiveness of these methods Zhu, Bernhard, and Gurevych manually annotated 120 sentence pairs from article pairs and found that sentence level TF-IDF outperformed other similarity methods. After application of TF-IDF based similarity, Zhu, Bernhard, and Gurevych were able to align 108K complex-simple sentence pairs. [86]

- **NEWSELA** [77]: This dataset was created because Xu, Callison-Burch, and Napoles found that Wikipedia is substandard data corpus for the following reasons: 1) Sentence extracted are prone to errors due to some drawbacks for sentence alignment methods 2) The data corpus has poor quality of simplifications. A manual analysis conducted by Xu, Callison-Burch, and Napoles on the widely used PKWP dataset [86] revealed that 50% of aligned sentence pairs are not simplifications. To overcome these problems Xu, Callison-Burch, and Napoles proposed a new corpus for Text Simplification called "NEWSELA". NEWSELA was created using news articles, specifically, 1130 news articles were collected and each of these articles was re-written 4 times by news editors on different grade level with Simp-4 denoting the most simplified level and Simp-1 denoting the least simplified level. These news articles were then used to extract

aligned sentence pairs by using Jaccard Similary measure [77].

There are many variants of these datasets available which are generated by using different text processing, text alignment techniques, etc.

Document level datasets

Most of the unstructured text data available on the internet(ex: research articles, news articles, etc) are documents or consist of multiple paragraphs. The TS models trained on sentence level TS tasks usually fail in such scenarios. Hence, the recent research has focused on the creation of documents of paragraph level datasets which can help in developement of ATS models which work on paragraph/document level data.

- **D-Wikipedia** [68]: The D-wikipedia dataset was created by aligning articles from EW and SEW. To create the dataset Sun, Jin, and Wan first downloaded dumps from official Wikipedia website and created over 170,000 article pairs. The authors removed articles which had more than 1000 words resulting in a dataset of 143,546 article pairs. The D-wikipedia dataset is further split into a training set containing 132K article pairs, a validation set containing 3K article pairs and remaining 8k article pairs for test set. [68]

- **20m Dataset** [25]: Similar to D-wikipedia dataset, 20m is also a document level dataset but it is designed for german language. Gonzales et al. extracted the dataset from Swiss news magazine 20 Minuten that consists of full articles paired with shortened, simplified summaries. The dataset does not have different simplification levels which were available similar to newsela. The corpus contains a total of 18,305 articles published since 2020. [25]

Evaluation Datasets

The datasets mentioned in earlier subsections are primarily used to finetune the models, but none of them provide a standard benchmark test set. To resolve this issue Xu et al. pioneered the first benchmark test set for TS models. This subsection gives details about the standard benchmark test sets used in the field of TS.

- **Turk Corpus** [78]: Xu et al. defined a novel metric called "SARI" and also curated a dataset (Turk Corpus) to calculate the SARI metric. To create Turk Corpus, the authors selected sentence pairs from PKWP dataset of similar length and were paraphrased only simplifications. Eight workers were hired from Amazon Mechanical Turk to write simplifications for selected normal English Wikipedia sentences without splitting and conserving information content as well as meaning. After the crowdsourcing operation the authors were able to gather a corpus 2350 sentences out of which 2000 sentences for tuning and 350 sentences for evaluation of models. In TS literature, Turk Corpus is a standard dataset used for evaluation of simplification models. [78]

- **Abstractive Sentence Simplification Evaluation and Tuning (ASSET)** [4]: Text Simplification involves several rewriting operations like replacing complex words, sentence splitting, removing irrelevant information, etc but the widely used Turk Corpus focuses on simplifications mostly created by paraphrasing. To address this issue, Alva-Manchego et al. curated a new dataset called ASSET which is made of several rewriting operations and can be used to evaluate TS models. To create ASSET, Alva-Manchego et al. used the same complex sentences from Turk Corpus [78] but crowdsourced the manual simplifications that encompass a broader set of rewriting operations. To accomplish this, Alva-Manchego et al. hired participants from Amazon Mechanical Turk. The participants were asked to solve Human Intelligence Tasks(HITs)

where each HIT had 4 normal sentences which needed to be simplified. Additionally, the participants were instructed to rate their simplifications on a Likert scale of 1-5. The authors finalized 10 simplifications per sentence resulting in a total 23950 human simplified sentences for the corresponding 2359 original sentences. [4]

2.3 Evaluation of Text Simplification Systems

Different evaluation strategies are suggested for evaluating outputs generated by text simplification systems. Broadly, they fall into two categories 1) Automatic Evaluation and 2) Human Evaluation. Most often, TS studies combine both evaluation strategies for output evaluation. This section highlights the methods used for automatic evaluations and human evaluations.

Human Evaluation

Due to the subjective nature of TS, especially when new text is generated, it has been recommended that human evaluations are the best approach to follow. The outputs are usually evaluated on three aspects: fluency, adequacy, and informativeness [69]. Fluency measures how well the text reads and ensures that there are no grammatical errors; adequacy measures whether the outputs convey the same meaning as the original text, and informativeness measures whether the outputs are able to capture important ideas present in the original text. These aspects are measured using the Likert scale, with a 1-5 scale or a 1-3 scale where a higher score denotes better simplification. Human evaluation has some disadvantages, with the major one being the requirement of native speakers with linguistic knowledge the evaluate the outputs generated by the TS system. In addition, humans are not consistent and have different opinion from one another, resulting in problems when comparing the outputs of

different systems. Moreover, human evaluation is very time-consuming and expensive. Thus researchers in the TS field are moving towards exploring and designing automatic evaluation metrics for TS systems.

Automatic Evaluations

One of the ways of performing automatic evaluations is via the help of readability indices and metrics used in Natural Language Generation (NLG) tasks. Readability indices tell how difficult is a piece of text to read. Some of the most commonly used readability indices are Flesch Kincaid Grade Level (FKGL) and Automatic Readability Index (ARI), while NLG metrics like SARI, ROUGE-1, ROUGE-2, etc are also widely adopted. This section gives a brief discussion on the mentioned metrics.

- Flesch Kincaid Grade Level (FKGL): Kincaid et al. [32] proposed the Flesch Kincaid Grade Level (FKGL), which gives a score that indicates a certain level that must be obtained to understand a particular text. A lower value of the FKGL score indicates that a particular text is simpler to read, and a higher score indicates that the text is complex [32]. FKGL is the most adopted measure of readability in text simplification literature. The FKGL for a text (S) is calculated using equation 2.1 [32]:

$$FKGL(S) = 0.38 \times \frac{\text{total words}}{\text{total sentences}} + 11.8 \times \frac{\text{total syllables}}{\text{total words}} - 15.59 \tag{2.1}$$

- Automatic Readability Index (ARI): Senter and Smith [62] developed ARI, which also measures readability just like FKGL. Just like FKGL, a lower ARI scores indicates that the text is easier to read and vice versa. For a text (S), the ARI score is denoted in the equation below

$$ARI(S) = 4.71 \times \frac{\text{total characters}}{\text{total words}} + 0.5 \times \frac{\text{total words}}{\text{total sentences}} - 21.43 \tag{2.2}$$

- ROUGE: Lin [37] proposed an automatic metric called "Recall Oriented Understudy for Gisting Evaluation" (ROUGE) for the task of evaluating text summarization models. Rouge scores are recall-based metrics and are computed by measuring the n-gram overlap between generated summary and the target summary. There are various variants of ROUGE scores like ROUGE-N, ROUGE-L, etc but the most used is ROUGE-N where N denotes the n-gram overlap between the reference and candidate summaries. [37]. ROUGE-N is given in the equation

$$\text{ROUGE-N} = \frac{\sum_{S \in \text{Reference Summaries}} \sum_{gram_n \in S} \text{Count}_{\text{match}}(\text{gram}_n)}{\sum_{S \in \text{Reference Summaries}} \sum_{gram_n \in S} \text{Count}(\text{gram}_n)} \tag{2.3}$$

- SARI: Xu et al. [78] proposed an automatic metric for evaluations of text simplification called "SARI" which uses F1-score of n-gram operations to measure simplicity. It computes an average of F1-scores for three n-gram operations: additions, keeps, and deletions, which are calculated based on the recall $R(n)$ and precision $P(n)$, based on the intersections of the input, output, and reference sets [78]. For each operation (i.e., add, keep, and deletion) F1-score is computed and SARI is the average of all the F1-scores as shown in Equation below:

$$
\begin{aligned}
\text{operation} &\in \{\text{add}, \text{keep}, \text{deletion}\} \\
P_{\text{operation}} &= \frac{1}{k} \sum_{n=[1,\cdots,k]} p_{operation}(n) \\
R_{\text{operation}} &= \frac{1}{k} \sum_{n=[1,\cdots,k]} r_{operation}(n) \\
F_{\text{operation}} &= \frac{2 \times P_{\text{operation}} \times R_{\text{operation}}}{P_{\text{operation}} + R_{\text{operation}}}
\end{aligned}
\tag{2.4}
$$

2.4 Language Models

Understanding important properties of text data using numerical representations is a challenging task and with the recent progress in research, language models have learned to do NLG tasks [55] like text summarization, simplification, question answering and semantic similarity [11, 12] quite efficiently. Initial research in the field of NLP leveraged standard seq2seq models like Long Short Term Memories (LSTM), Gated Recurrent Units (GRU), and Recurrent Neural Networks (RNN) due to their impressive performance on NLG tasks, but the recent focus has shifted to the applications of transformer [35] based language models to NLG tasks. The research in the field of TS has also benefited from the adaptation of transformer-based models. This section highlights a few important transformer-based models used in the field of TS.

Generative pre-training

The amount of large unlabelled text corpora is abundant as compared to the amount of labelled corpora for downstream tasks. Radford et al. demonstrated impressive results on downstream tasks by generative pre-training of language model on diverse text corpora, then

finetuning the model on downstream tasks. Their pre-training process consists of two stages which are as follows:

- Unsupervised pre-training: Given a corpus of tokens $U = \{u_1, u_2, \cdots, u_n\}$, the model is trained using standard language modeling objective to maximize the likelihood of generating the next token given the previous tokens and is calculated as depicted in equation

$$L(U) = \sum_i logP(u_i|u_{i-k}, \cdots, u_{i-1}; \theta) \tag{2.5}$$

  where k is the context window size, and P is the conditional probability modeled using a neural network with parameters θ. [57]

- Supervised finetuning: Once the model is pre-trained in an unsupervised fashion, it is then finetuned on a downstream task via the help of a labeled dataset. The downstream task could be any standard NLG tasks, for example, text classification, text summarization, sentiment analysis, etc.

Unlike the standard transformer encoder-decoder architecture which is generally used of NLG tasks, Radford et al. used only the decoder portion of the transformer model. Specifically, their architecture consisted on 12 transformer decoder layers followed by a linear head whose architecture is dependent on the downstream task. Radford et al. trained the model on diverse unlabeled corpora and later finetuned the model for downstream tasks achieving state-of-the-art results on Natural Language Inference (NLI) and Question Answering Tasks (QA). Additional details about the model architecture and adaptation to other tasks is available in [57].

BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension

Lewis et al. introduced BART, a sequence-to-sequence model based on the transformer-based NMT architecture. BART is pre-trained using two methods 1) text corruption via the help of a noising function and 2) learning a model to generate the original text from the corrupted text. This pre-training scheme has allowed the model to achieve state-of-the-art results on many NLG tasks, including text summarization and machine translation [35]. BART has a sequence-to-sequence architecture, with both the encoder and decoder having six layers each. The model is pre-trained using various noising schemes, which are mentioned below.

- Token Masking: Similar to BERT [31], random tokens are selected and replaced with <MASK> token.
- Token Deletion: Random tokens are deleted from the inputs.
- Text infilling: Random number of text spans are selected and replaced with <MASK> token similar to SpanBERT [29].
- Sentence Permutation: A document is divided into sentences by splitting it on full stop. Then these sentences are shuffled randomly.
- Document Rotation: A token is chosen randomly from the document and then the document is rotated so that the selected token is the start token. This task aims to teach the model how to identify the start of documents.

BART models which are pretrained using text infilling tasks have shown better performance on various standard NLG benchmarks like SQUAD, XSUM, CONVAI as compared

to other pre-training regimes. For more information about results, training process and abalation studies, the readers are suggested to refer to [35].

PEGASUS: Pre-training with Extracted Gap-sentences for Abstractive Summarization

Zhang et al. proposed PEGASUS, a transformer-based language model for the purpose of text summarization with a specific focus on abstractive summarization. PEGASUS was pre-trained using a novel training method called "Gap Sentence Generation" (GSG). GSG works by masking whole sentences from the document and forces the model to generate the masked sentences from the rest of the documents. The authors used three strategies to select the gap sentences (i.e., the sentences that will be replaced by a <MASK> token). The strategies used are as follows.

- Random: Select m sentences as random.
- Lead: Select first m sentences.
- Principal: Select $top - m$ scored sentences via some importance scores. Zhang et al. used the ROUGE-1 score as the importance metric, where the rouge score is calculated between the sentence and the remaining document.

The experiments conducted by Zhang et al. revealed that pre-training using the GSG method helped them achieve state-of-the-art or similar results on 12 benchmark abstractive summarization datasets. Additional details about the benchmarks used for testing can be found in [82].

Chapter 3

Bibliometric Analysis of the Text Simplification Literature

All of this chapter is submitted at a reputed journal as:

- Phatak, A., Liyanege, R., & Mago, V. (2022). Bibliometric Analysis of the Text Simplification Literature

Over the course of my degree, I researched topics related to TS to expand my knowledge in the field. As a result, I performed Biblometric analysis to understand and analyze different research techniques and research applications of TS.

3.1 Introduction

Text Simplification (TS) is a subdomain of natural language processing (NLP) that applies computational techniques to convert complex text into a simpler version, thus making it readable and understandable to a wider audience ranging from people with learning disabilities to making research accessible to general audience [70, 47, 77, 63]. With all of these applications, TS has become a significant research area that warrants a thorough investigation of the literature to reveal the domain's emerging trends and research constituents.

Reviewing the academic literature is important primarily to gather existing findings and identify gaps within a research context. Nevertheless, the investigation of the internal structure of the scientific literature, which consists of entities such as papers, topics, authors, publishers, and affiliations, is also advantageous for researchers to understand the development of a particular research direction [19, 36]. This structural exploration of the literature can be accomplished through a quantitative approach called Bibliometric Analysis [19]. Moreover, performing bibliometric analysis over other common literature review techniques, such as systematic review and meta-analysis, is important as it analyzes the relationships between these different entities of publications [19]. This relational analysis helps to uncover useful information, for example, the contribution of authors and organizations to specific topics, influential studies in a domain and the connections between them, and the trends of a particular research field [19, 56, 1]. Based on these considerations, this study conducts a bibliometric analysis of the papers and meta-data collected from the area of text simplification from 2012 to 2022. Specifically, we constructed our work toward answering six research questions which are as follows:

- RQ1: How has the domain of TS evolved over years?

- RQ2: Which authors contribute most to the research in the field of TS?
- RQ3. Which countries contribute most to the TS research?
- RQ4. How have researchers collaborated within the field of TS?
- RQ5. How have the collaborations developed over time?
- RQ6. What are the trends of TS research?

By addressing RQ1, we evaluate the progress of the field of TS. The next two research questions uncover the leading authors and countries contributing to the domain. While RQ4 identifies the collaborations between the authors of the network, RQ5 demonstrates how those collaborations have evolved over years. Finally, RQ6 reveals the emerging topics within the TS context.

The rest of the paper is organized as follows: the Related works section, provides a brief review of the literature on TS and bibliometric analysis in the field of NLP. Next, the methodology section explains the techniques and approaches we used to perform the bibliometric analysis, including how the data collection and analysis were performed. The results of the study and further discussion are presented in the Results section. Finally, we conclude and remark on future research directions.[1]

3.2 Related Work

Although the main purpose of this study is to perform a bibliometric analysis in the field of text simplification, it is also worthwhile to familiarize with the background of the domain. Hence, in this section, we will briefly discuss the context of TS, including its applications

[1]The data and code is available on github

and potential techniques. Moreover, we will discuss techniques and findings retrieved from the literature on the bibliometric analysis of both NLP and TS.

Text Simplification

The process of text simplification is comprised of a few other techniques, such as complex word replacement to identify and replace difficult words with their simpler forms [30], elaborative simplification to expand upon the main ideas of text [66], and text summarization to make the content of the text concise [64]. These techniques are proven to be very effective and are usually used in conjunction with other techniques to improve comprehensibility for readers.

Early studies in the field of TS have mainly focused on the lexical simplification technique. A lexical simplification system works on replacing complex words with simpler alternatives taken from lexical databases, such as Paraphrase Database (PPDB) [22] and WordNet [71]. More recently, the current research has progressed from lexical simplification to building Automated Text Simplification (ATS), an approach pioneered by Nisioi et al. [45] to leverage the essence of deep learning-based NLP techniques in automating the process. Although most of the research in the field of ATS is supervised in nature, which requires paired datasets of complex and simple text, the development of self-supervised or similarly unsupervised techniques is also significant. Currently, transformer-based supervised methods, including Deep Memory Augmented Sentence Simplification (DMASS) [84], and AudienCe-CEntric Sentence Simplification (ACCESS) [40] have achieved state-of-the-art results on standard TS datasets.

The applicability of TS has evolved remarkably among diverse target audiences with specific application focuses [6]. One of the most prominent beneficiaries has been second

language learners, for whom various approaches to simplifying text have been proposed. These approaches often focus on sentence-level simplification [38]. Moreover, TS also assists people with learning disorders such as dyslexia [60] and autism [20] or linguistic impairments by reducing the syntactic complexity of natural languages. Novice readers (both children and adults) have also benefited from TS through both syntactic and lexical simplification [16].

Biblometric Analysis

Bibliometric analysis is a method for analyzing and exploring large amounts of scientific data available in academic literature [19]. This analysis reveals emerging areas in a specific field and enables researchers to understand how a particular field is evolving over time.

Chen et al. [14] investigated publications that applied NLP in the medical domain. They collected data from PubMed between 2007 and 2016 and used bibliometric techniques to analyze and understand important medical research topics, scientific collaborations between affiliations and authors, and how NLP-empowered medical research has been growing over the years. Another study conducted a bibliometric analysis of research publications retrieved from the Association of Computational Linguistics (ACL) and Empirical Method in Natural Language Processing (EMNLP) conferences [24]. The analysis of papers across two decades revealed significant topics from both conferences, as well as how the topical focus of both conferences differed and evolved over time. Similarly, Radev et al. [56] conducted a thorough bibliometric and network analysis of papers published in ACL. They extracted paper and author citation data from publications and analyzed them through networks to identify the most central papers and authors. The researchers also quantified the analysis with network statistics.

Studies on Bibliometric Analysis of TS were extremely rare in the literature, with the exception of Özcan and Batur [47] who explored eight research questions. However, their dataset was bound to journal articles published on Scopus databases under the social science domain. As a result, although they collected data from 1975 to 2020, the size of their dataset was limited to 194 articles. Therefore, a comprehensive bibliometric study on TS, performed on a considerable amount of data that is not restricted to a specific application domain or database, is necessary. Considering the importance of the field and the lack of existing knowledge on bibliometric information, this study conducted further investigations on the TS domain. Our findings will enrich the bibliometric information and help to assess the research networks and directions in the field of TS.

3.3 Methodology

Data Collection

Google Scholar (GS) is the largest web search engine for academic literature. Using GS, researchers can access metadata associated with a research article, such as citation counts, authors, years, and venues of publications. Additionally, information about the researchers themselves is also available, including their papers, h-index, and citations, as well as other researchers with whom they have published. Due to this richness of data, we were able to collect both articles and their associated metadata from GS using the Publish-or-Perish software [27], focusing on articles published between 2001 and 2022. We used the keyword "Text Simplification" to search research article titles, which resulted in 656 articles. We also extracted authors' h-indexes, total citations, and affiliations from their GS profiles using Scholarly API.

Network Graph Generation

Investigation of social structures can be conducted through Social Network Analysis, a technique that constructs graphs to represent the underlying associations between entities of a social network [23]. This approach can be used in bibliometric analysis to denote the relationships between several types of networks, including co-authorships, affiliations and citations [56, 14]. In this study, we generated network graphs to understand how the research community collaborates in the field of TS.

Collaboration Analysis

The analysis of collaborations reveals how members of the research community are sharing their expertise to advance research in the field of TS. A co-authorship network is a common type of collaboration network, where the nodes in the network graph represent the researchers and the edges between them represent the strength of collaborations among them [21]. For this study, we generated a network graph that depicts author-author collaboration and the data required to understand these collaborations; authors and their co-authors were extracted from GS.

Algorithm 1: Collaboration Generation

Input: D: Dataset
Output: G: Author network
Variables: $Authors$: The unique authors in dataset, $CoAuthorData$: The collected co-authors for all unique authors in the dataset.

```
/* Method to extract unique authors in dataset.                    */
```
1 $Authors \leftarrow ExtractUniqueAuthors(D)$;
```
/* Method collects all co-authors                                  */
```
2 $CoAuthorData \leftarrow CollectCoAuthors(Authors)$;
```
/* Method builds collaboration graph from using the Authors and
   Co-Authors.                                                     */
```
3 $G \leftarrow BuildCollaborationGraph(Authors, CoAuthorData)$ **return** G ;

As illustrated in Algorithm 1, the following steps were performed to create the network graphs representing the collaboration between authors. First, the function *ExtractUniqueAuthors* extracts all the unique authors available in the dataset (D). Next, the *CollectCoAuthors* function was developed to find the co-authors from the extracted unique authors' GS profiles using Scholarly API. Finally, these collected data were used to create a network graph using the *BuildCollaborationGraph* function, which is further described in Algorithm 2. This function requires two arguments – the unique authors in the dataset (Authors) and co-authors of all the unique authors (*CoAuthorData*). *CoAuthorData* is a hashmap with author as keys and their corresponding coauthors as values. As the output, an undirected network is generated using the *networkX* module in Python and it is visualized using the *d3js* library in Javascript.

Algorithm 2: Build Collaboration Graph

Input: *Authors*: The unique authors in dataset, *CoAuthorData*: The collected co-authors for all unique authors in the dataset.

Output: *G*:Author network

```
1 Function BuildCollaborationGraph(Authors, CoAuthorData)
      /* Create empty graph using networkx(nx) module in python.   */
2     G ← nx.Graph() ;
3     for author in Author do
          /* Lookup coauthors in CoAuthorData dictionary            */
4         coauthors ← CoAuthorData[author] ;
5         for coauthor in coauthors do
6             G ← G.AddEdge(author, coauthor)
7         end
8     end
9     return G
```

Temporal Analysis

While collaboration analysis gives us insight into how authors are collaborating, the temporal analysis highlights how these collaborations have been developed over the years. However, as the GS does not provide year-wise information about the association between authors and co-authors, it is unknown when a particular author has collaborated with another author. To overcome this limitation, we performed a temporal analysis by creating citation networks between authors. Considering a span of five years, this study generated these citation networks for each year segment from ≪2012-2017≫ to ≪2016-2021≫ to understand the evolution of collaborations in the field of TS during the years 2012 to 2021.

The process of data collection for temporal analysis consists of two important steps: 1) root paper selection and 2) cited paper collection. The root paper selection filters the most cited paper from the dataset for a particular base year from 2012 to 2016. The cited paper collection involves collecting the cited papers within the relevant five-year segment using Publish-Perish software. We collected these cited papers in two sets, where the first set consisted of papers that have cited the selected root paper (say S1), and the second set included the papers that have cited the papers in the first set. The data collected for each year segment were separately stored in JSON files; for example, all the data for the year 2015 that was collected through ≪2015-2020≫ can be viewed github.

The dataset in the JSON file is structured as key-value pairs, where the keys are the titles of papers in S1. The values are represented in a nested list of sublists, where for each key, the first level of sublists contains the papers that cited it and the second level of sublists consists of the authors of each of these cited papers. Finally, this collected dataset was used to generate the citation networks using the strategy described in Algorithm 3.

The function *readJson* first reads the JSON data (CitationDataset) and stores them in a

variable (citationData). The next steps (line 2 - line 9) iterate over these data to extract the authors who cited a particular author (citedAuthors) and add them to an empty dictionary (AuthorList) with the author as a key and the cited authors as the values. Finally, this *AuthorList* is used to generate the undirected citation graph using the *networkX* module. We followed this method to find the connections of all unique authors in the *CitationDataset* and this resulted in an undirected graph where the authors and collaborations are represented by nodes and their edges respectively. The size of the node was decided by the number of collaborations associated with it. By following the same mechanism for each five-year span, we were able to discover how the collaborations of authors in TS developed over the years.

Topic Modelling and Word Frequency Analysis

Word Cloud is a way of visually representing text data in the form of unigrams or tags where the importance of single words is indicated by their size and color [14]. Similarly, topic modeling is a technique to extract the thematic structure of documents by analyzing the words from the original sources [39]. We developed word clouds for the abstracts of publications to collect essential keywords in the domain of TS and to gain a fundamental understanding of how TS research has been applied. Moreover, we conducted topic modeling on the abstracts to generate a detailed overview of the field of TS by discovering its important themes as described in Algorithm 4.

In Algorithm 4, the function *CleanAndTokenizeData* first cleans the input abstracts (D) by removing all non-alphabets (punctuation, numbers, new-line characters, and extra spaces) and URLs, and then uses *Natural Langauge ToolKit (NLTK)* tokenizer to tokenize the data. Next, the function *RemoveStopWords* removes the stopwords from D based on the identified *STOPWORD* list. The cleaned data is then used to train the *BERTTopic* model (T), which is a topic modeling technique that clusters the keywords into different topics using

Algorithm 3: Citation Graph Generation

Input: *citationDataset*: JSON output from publish-perish software for a particular year.
Output: G: Author network
Variables: *citationData*: Dictionary containing papers with authors and their cited papers, *AuthorList*: An empty dictionary that will store author and their corresponding co-authors

```
// Method to read json file
citationData ← readJson(citationDataset) /* Iterate over the json data and
    store the authors and the co-authors                                  */
for paperTitle in citationData do
    paperData ← citationData[paperTitle] ;
    for (citedPaper, citedAuthors) in paperData do
        for author in citedAuthors do
            AuthorList[author].add(citedAuthors)
        end
    end
end
// Create empty networkx(nx) graph.
G ← nx.Graph() ;
/* Iterate over adjacency list and add edges to the graph.              */
for author in AuthorList do
    citingAuthors ← AuthorList[author] ;
    for citingAuthor in citingAuthors do
        G ← G.AddEdge(author, citingAuthor);
    end
end
return G;
```

Algorithm 4: Topic Modelling Steps

Input: D: Paper Abstracts, STOPWORD: list of stopwords, T : Topic Model
Output: $T_{trained}$: Trained Topic Model

$D \leftarrow CleanAndTokenizeData(D)$;
$D \leftarrow RemoveStopWords(D, STOPWORD)$;
$T_{trained} \leftarrow TrainBertTopic(D)$;
$clusters \leftarrow getTopics(T_{trained})$;
return *clusters*

transformers and c-TF-IDF [26]. Next, the function *getTopics* outputs the set of keywords for each topic. Finally, with the help of domain experts, the keywords grouped under each topic were used to assign a suitable title for the cluster.

3.4 Results and Discussion

In this section, we present the findings for each research question along with further discussion for knowledge extraction.

RQ1: How has the domain of TS evolved over years?

The development of the TS field is represented by the number of papers published per year from 2001 to 2022; these findings are shown in Figure 3.1. Overall, the number of publications has increased over time, particularly since 2010. Significantly, approximately 25 percent of total publications occurred in 2020 and 2021, with the highest number of publications in 2021 (about 85).

Meanwhile, it is noteworthy that this overall upward trend has been interrupted by several declines in publication counts during 2015, 2017, 2019, and 2022 when compared to the preceding years. A possible reason for this pattern could be that the prominent conferences in this area occur only once every two years. Additionally, this decrement can be ignored in 2022 as the data collection period ended in November, and several papers could have been under review by this point. The overall trend can be further described by the average number of papers published in different noticeable groups of years; $\ll$2001-2009$\gg$, $\ll$2010-2015$\gg$, and $\ll$2016-2022$\gg$ where it is 7, 27, and 60 respectively. In conclusion, the total growth in the number of publications indicates that, over time, the field of TS has gained traction in the research community.

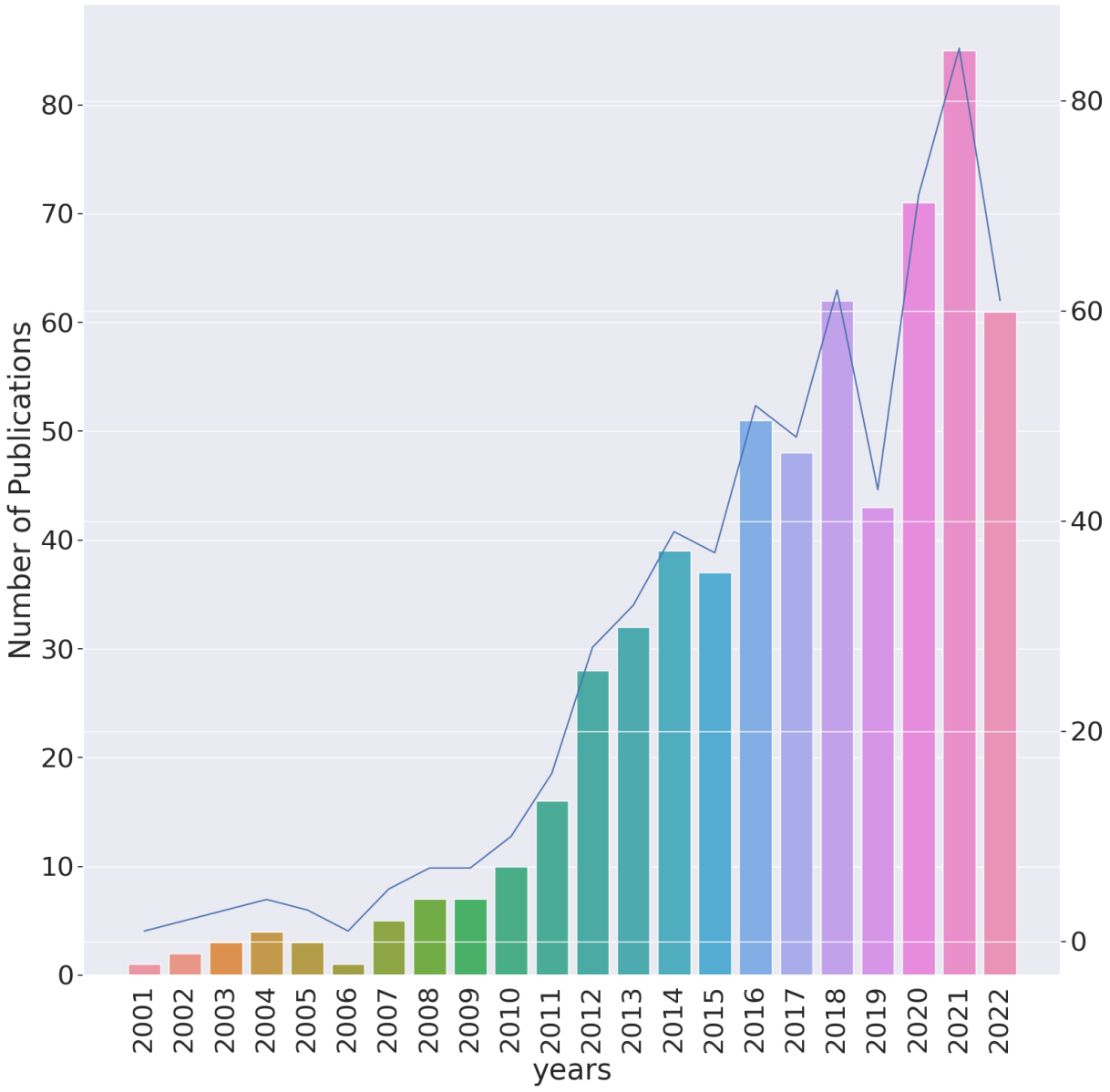

Figure 3.1: Distribution of the number of publications in the field of TS over the years

RQ2: Which authors contribute most to the research in the field of TS?

We analyzed publications based on the engagement of authors in TS research, where the most influential authors were determined by a citation count. The ten most influential authors and their details are recorded in Table 3.1. From these recorded data, the top author, Qiang Yang, has more than 85k citations while all other authors have citation counts between 30k-65k. Moreover, although seven out of ten authors are from universities, the most influential author represents a Chinese company called WeBank. Additionally, 32 percent of the total top-10 citations are from authors in industry. This finding highlights the fact that the field of TS has equally grasped the attention of authors from both industry and academia.

Rank	Author	Citations	Affiliation
1	Qiang Yang	86146	WeBank
2	David N. Kennedy	62767	University of Massachusetts
3	Ricardo Baeza-Yates	53154	University of Pompeu Fabra
4	Cathy Wu	51981	University of Delaware
5	Allan Peter Davis	44228	North Carolina State University
6	Xindong Wu	39291	Hefei University of Technology
7	Danielle McNamara	38123	Arizona State University
8	Marti A Hearst	37344	University of California
9	Antoine Bordes	35481	Meta-AI
10	Kevin Knight	31861	DiDi Labs

Table 3.1: Top 10 influential authors with their citations and affiliations

RQ3: Which countries contribute most to the research?

Awareness of the country-wise contribution to the research context is important for researchers to identify the locations where TS research is prominent. The country-wise influ-

Rank	Country	Number of Publications
1	USA	94
2	UK	47
3	Germany	34
4	Spain	20
5	France	18
6	India	16
7	Belgium	16
8	Canada	14
9	Japan	14
10	Italy	13
11	China	12
12	Brazil	12
13	Switzerland	11

Table 3.2: Top 10 influential countries with their corresponding number of publications

ence on TS was investigated by determining the location (country) of the first authors in publications. However, as GS does not provide this information, we have manually extracted it from the authors' affiliations indicated in papers collected from 2001 to 2022.

Table 3.2 shows the top countries with more than 10 publications in TS. According to these findings, the United States of America is the prominent contributor with around 33 percent of the total papers published by the top-ten countries. However, the majority of the individual contributors are from Europe, with England, Germany, France, and Spain as next four highly ranked countries.

RQ4: How have researchers collaborated in the field of TS?

The study investigated the extent of cooperation among researchers in the field of TS by constructing an author-author network graph. Figure 2 depicts a sub-network that showcases the connections of Ricardo Baeza-Yates from the University of Pompeu Fabra in Spain who

has collaborated with the highest number of authors in the field of TS, with a count of 177. He has diverse collaborations with authors from both academia and industry, such as Facebook, Amazon, and Spotify, emphasizing the multidisciplinary nature of cooperation in the field. The overall network comprises 3,945 nodes that represent the authors who have made significant contributions to the field of TS from 2001 to 2022. The connecting links signify the co-authorship collaborations that have facilitated the advancement of the field. For those interested in further exploration, higher resolution graphs are available at https://bblts.datalab.science/.

RQ5: How have the collaborations developed over time?

We performed a temporal analysis to evaluate the evolution of the collaboration among authors in the TS context. This collaboration analysis was conducted based on the extracted citation relationships within different five-year segments from 2012 to 2021; we did not include citations from 2001 to 2011 as the total number of publications during that period is not significant. For each of the included time segments, we generated a network graph representing the citation connections between authors and two bar plots displaying the top five 'author-citation' and 'author-affiliation' distributions. Figure 3.3 shows these bar plots for the first and last year segments, ≪2012-2017≫ and ≪2016-2021≫. Overall, the number of authors and citations has increased significantly, with the highest number of authors and citation counts in the period from ≪2016-2021≫. Moreover, it is noteworthy that the top-five-most reputed affiliations representing authors have changed with time, where in ≪2012-2017≫ all were universities, and in ≪2016-2021≫ four out of five are from industry. Among the reputed industrial affiliations engaged with TS research, Google and Microsoft ranked at the top for many years. In addition, Amazon, IBM, and Facebook are

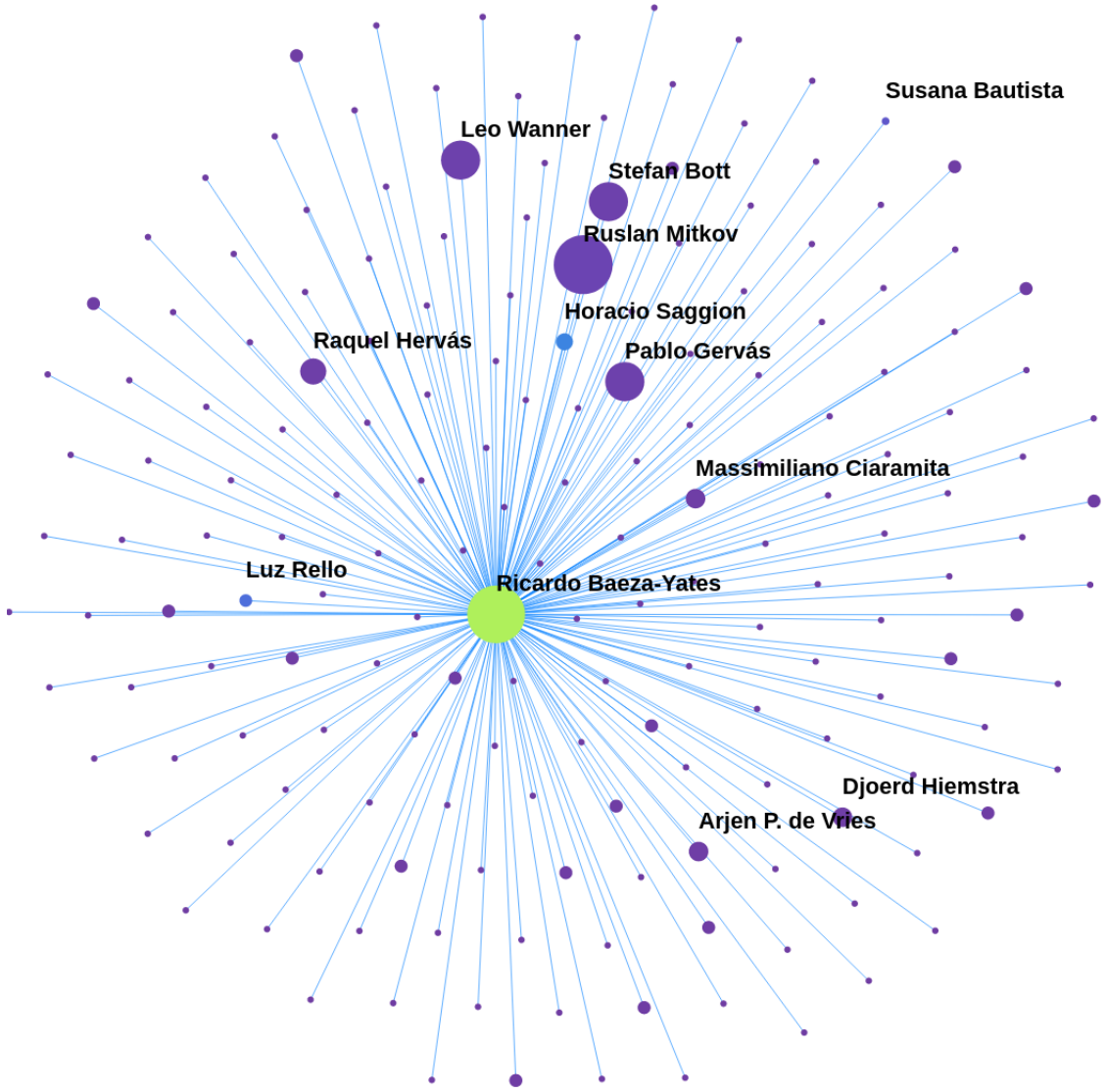

Figure 3.2: Network graph for author with the highest number of connections. Higher resolution graphs are available here

recognized as significant IT companies in the TS domain.

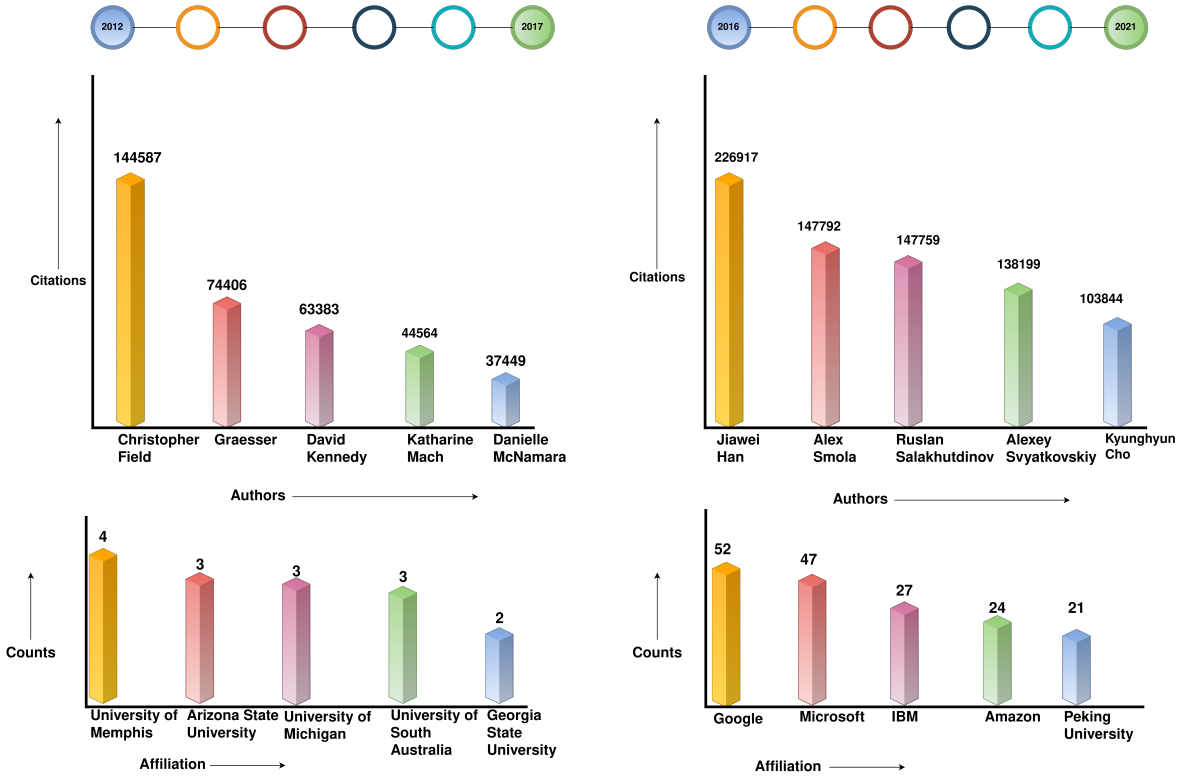

Figure 3.3: Citations of influential authors and the number of authors in dominant affiliations for the year segment ≪2012-2017≫ and ≪2016-2021≫

RQ6: What are the trends of TS research?

The research trends in the domain of TS are represented in Figure 3.4, which visualizes the results of word cloud analysis conducted on the publication abstracts. This investigation revealed some important keywords related to different application domains, such as "medical", "lexical", "readability", "linguistic", "Spanish" and "summarization". This finding indicates that TS research has been applied in various domains ranging from medicine to computer science and literacy.

Additionally, the set of keywords extracted from these abstracts was used in topic modeling. As shown in Table 3.3, we identified important groups of topics based on the list of

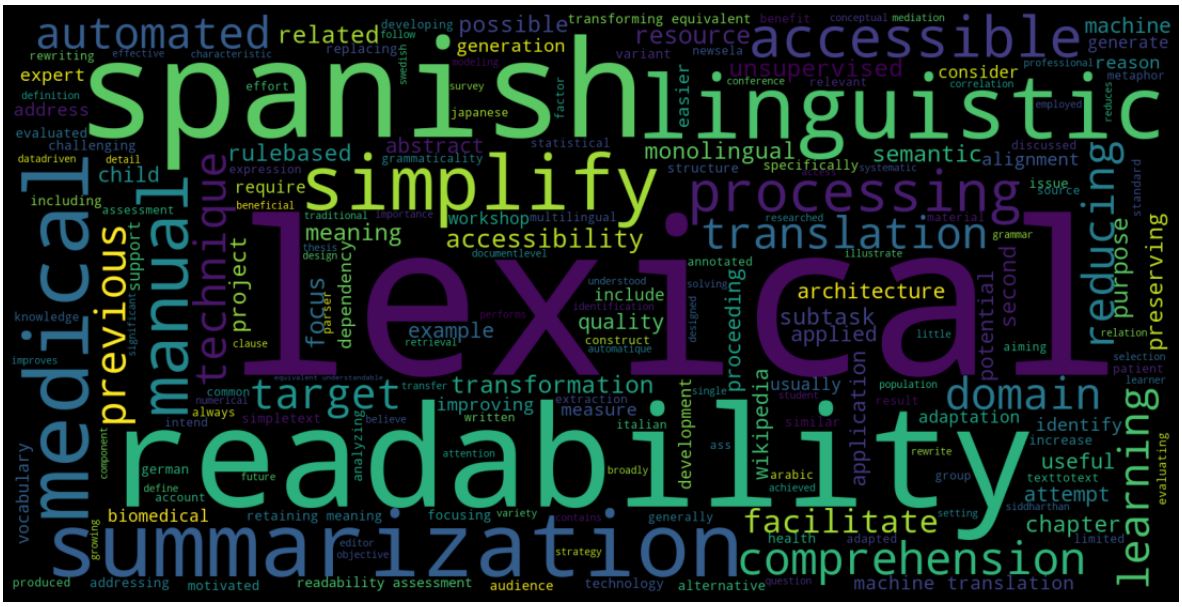

Figure 3.4: Wordcloud Analysis of Abstracts

keywords classified under them, with the first keyword of each topic being assigned with higher confidence. According to this analysis, application of the research conducted on TS has been most popular within medical, multilingual, and NLP areas.

Topics	Keywords
Text simplification applications in healthcare	medical, biomedical, health, patient
Multilingual text simplification for non-native readers	Spanish, Italian, adaptation, support, production
Text simplification applications in NLP	lexical, readability, translation, summarization, linguistic

Table 3.3: Topics of TS research and the common set of keywords

3.5 Conclusion

Text Simplification is a widely applied NLP technique that improves text comprehensibility for people with limited literacy. Due to the high demand and vast applicability of the field, researchers can benefit from a thorough analysis and understanding of the development of research in TS. In this paper, we investigated the bibliometric information of TS literature from 2001 to 2022 by addressing six research questions. First, we explored the evolution of the domain of TS over the years. The increasing number of publications in the community indicated the increasing interest in the field. Second, we determined the most influential authors within this domain, which revealed the equal engagement of both industry and academia in TS research. Third, we identified the United States and numerous European countries as the most significant contributors to the TS literature. Fourth, we investigated the collaboration among researchers in the field of TS through a network of authorships. Fifth, the evolution of collaborations among researchers was extracted through citation trees developed for different year spans. These connections provide further evidence of how the contribution of industry and academia in the TS domain has changed over time. Finally, we investigated the directions of TS research through wordcloud and topic modeling techniques performed on publication abstracts. This analysis indicated that the main applications of TS research occur in healthcare, multilingual TS for non-native readers, and NLP. The outcomes of this study are useful for gaining familiarity with the structural relationships and trends within the literature of the TS field.

Limitations

The main constraints in this study were the size of the dataset and the amount of manual work performed during the extraction of some data. This study could be further expanded by inculcating other data sources, such as the Web of Science and Semantic Scholar. Moreover, the papers can be further analyzed by extracting more textual information to uncover additional insights about research being conducted in the field of TS.

Chapter 4

Medical Text Simplification Using Reinforcement Learning (TESLEA): Deep Learning–Based Text Simplification Approach

All of this chapter was published in the following peer-reviewed journal [52]:
• Phatak A, Savage D, Ohle R, Smith J, Mago V Medical Text Simplification Using Reinforcement Learning (TESLEA): Deep Learning–Based Text Simplification Approach. JMIR Med Inform 2022;10(11):e38095.

Using the advancements in the field of Natural Language Processing, this chapter proposes an approach to automate Text Simplification for medical text data.

4.1 Introduction

Bio-medicine research is a vital source of information on new clinical trials for drugs and treatments for a wide range of diseases. This information is publicly available but it is often filled with complex medical terminologies, making it hard for the general public to comprehend. To solve this issue, one solution is to manually simplify the language used in biomedical texts so that it can be understood by a wider audience. Although manual text simplification is effective, it cannot cope with the rapidly expanding volume of biomedical literature. Therefore, there is a pressing need to develop NLP approaches that can automatically simplify biomedical texts, making them more accessible to the public.

The aim of this study is to develop an automatic TS approach that is capable of simplifying medical text data at a paragraph level, with the goal of providing greater accessibility of biomedical research. This paper uses RL-based training to directly optimize 2 properties of simplified text: relevance and simplicity. Relevance is defined as simplified text that retains salient and semantic information from the original article. Simplicity is defined as simplified text that is easy to understand and lexically simple. These 2 properties are optimized using TS-specific rewards, resulting in a system that outperforms previous baselines on Flesch-Kincaid scores. Extensive human evaluations are conducted with the help of domain experts to judge the quality of the generated text.

The remainder of the paper is organized as follows: The "Related Works" section provides brief information on models, datasets and evaluation metrics in TS. The "Methods" section provides details on the data set, the training procedure, and the proposed model, and describes how automatic and human evaluations were conducted to analyze the outputs generated by the proposed model (TESLEA). The "Results" section provides a brief description of the baseline models and the results obtained by conducting automatic and manual

evaluation of the generated text. Finally under the "Discussion" section, we highlight the limitations, future work, and draw conclusions. [1]

4.2 Related Work

Text Simplification Approaches

Initial research in the field of TS focused on lexical simplification (LS) [10, 48]. An LS system typically involves replacing complex words with their simpler alternatives using lexical databases, such as the Paraphrase Database [22], WordNet [71], or using language models, such as bidirectional encoder representations from transformers (BERT) [54]. Recent research defines TS as a sequence-to-sequence (seq2seq) task and has approached it by leveraging model architectures from other seq2seq tasks such as machine translation and text summarization [86, 76, 18]. Nisioi et al. [45] proposed a neural seq2seq model, which used long short-term memories (LSTMs) for automatic TS. It was trained on simple-complex sentence pairs and showed through human evaluations that the TS system–generated outputs ultimately preserved meaning and were grammatically correct [45]. Afzal et al. [2] incorporated LSTMs to create a quality-aware text summarization system for medical data. Zhang and Lapata [83] developed an LSTM-based neural encoder-decoder TS model and trained it using reinforcement learning (RL) to optimize SARI directly [78] scores along with a few other rewards. SARI is a widely used metric for the automatic evaluation of TS.

With the recent progress in natural language processing research, LSTM-based models were outperformed by transformer-based language models [73, 35, 57]. Transformers follow an encoder-decoder structure with both the encoder and decoder made up of L identical

[1] The data and code is available on github

layers. Each layer consists of 2 sublayers, one being a feed-forward layer and the other a multi-head attention layer. Transformer-based language models, such as BART [35], generative pretraining transformer (GPT) [57], and text-to-text-transfer-transformer [58], have achieved strong performance on natural language generation tasks such as text summarization and machine translation.

Building on the success of transformer-based language models, recently Martin et al. [41] introduced multilingual unsupervised sentence simplification (MUSS) [41], a BART [35] based language model, which achieved state-of-the-art performance on TS benchmarks by training on paraphrases mined from CCNet [74] corpus. Zhao et al. [85] proposed a semisupervised approach that incorporated the back-translation architecture along with denoising autoencoders for the purpose of automatic TS. Unsupervised TS is also an active area of research but has been primarily limited to LS. However, in a recent study, Surya et al. [69] proposed an unsupervised approach to perform TS at both the lexical and syntactic levels. In general, research in the field of TS has been focused mostly on sentence-level simplification. However, Sun, Jin, and Wan [68] proposed a document-level data set (D-wikipedia) and baseline models to perform document-level simplification. Similarly, Devaraj et al. [18] proposed a BART [35]-based model that was trained using unlikelihood loss for the purpose of paragraph-level medical TS. Although their training regime penalizes the terms considered "jargon" and increases the readability, the generated text has lower quality and diversity [18]. Thus, the lack of document or paragraph-level simplification makes this an essential work in the advancement of the field.

Text Simplification Datasets

The majority of TS research uses data extracted from Wikipedia and news articles [83, 15, 28]. These data sets are paired sentence-level data sets (i.e., for each complex sentence, there is a corresponding simple sentence). TS systems have heavily relied on sentence-level data sets, extracted from regular and simple English Wikipedia, such as WikiLarge [83], because they are publicly available. It was later shown by Xu, Callison-Burch, and Napoles [77] that there are data quality issues in the data sets extracted from Wikipedia. They proposed the Newsela corpus, which was created by educators who rewrote news articles for different school-grade levels. Recent work has focused on the construction of document-level simplification data sets [41, 68, 34]. Sun, Jin, and Wan [68] constructed a document-level data set, called D-Wikipedia, by aligning the English Wikipedia and Simple English Wikipedia spanning 143,546 article pairs. Although there are many data sets available for sentence-level TS, data sets for domain-specific paragraph-level TS are lacking. In the field of medical TS, Štajner [67] constructed a sentence-level simplification data set using sentence alignment methods. Recently, Devaraj et al. [18] proposed the first paragraph-level medical simplification data set, containing 4459 simple-complex pairs of text, and this is the data set used for the analysis and baseline training in this study. A snippet of a complex paragraph and its simplified version from the data set proposed by Devaraj et al. [18] is shown in Figure 4.1. The data set is open-sourced and publicly available [18].

Text Simplification Evaluation

The evaluation of TS usually falls into two categories: automatic evaluations and manual (i.e., human) evaluations. Because of the subjective nature of TS, human evaluations is still considered the best option for evaluating TS systems [69]. Automatic evaluation metrics

COMPLEX MEDICAL PARAGRAPH

Two studies enrolled preterm infants with respiratory distress. Amato (1988) allocated infants to L-thyroxine $50\mu g/dose$ at 1 and at 24 hours or no treatment. Amato (1989) allocated infants to L-triiodothyronine $50\mu g/day$ in two divided doses for two days or no treatment. Both studies had methodological concerns including quasi-random methods of patient allocation, no blinding of treatment or measurement and substantial post allocation losses. Neither study reported any significant benefits in neonatal morbidity or mortality from use of thyroid hormones. Meta-analysis of two studies (80 infants) found no significant difference in mortality to discharge (typical RR 1.00, 95% CI 0.47, 2.14). Amato 1988 reported no significant difference in use of mechanical ventilation (RR 0.64, 95% CI 0.38, 1.09). No significant effects were found in use of mechanical ventilation, duration of mechanical ventilation, air leak, CLD at 28 days in survivors, patent ductus arteriosus, intraventricular haemorrhage or necrotising enterocolitis. Neurodevelopment was not reported. There is no evidence from controlled clinical trials that postnatal thyroid hormone treatment reduces the severity of respiratory distress syndrome, neonatal morbidity or mortality in preterm infants with respiratory distress syndrome.

SIMPLE MEDICAL PARAGRAPH

This review found two small trials that compared the use of thyroid hormones to no treatment in infants with breathing problems in the first hours after birth. No benefit was found from use of these hormones on severity of breathing problems or complications that occurred as a result of these breathing problems. The effect on longer term development was not reported.

Figure 4.1: Complex medical paragraph and the corresponding simple medical paragraph from the dataset [17]

most commonly used include readability indices such as Flesch-Kincaid Reading Ease [32], Flesch-Kincaid Grade Level (FKGL) [32], Automated Readability Index (ARI), Coleman-Liau index, and metrics for natural language generation tasks such as SARI [78] and BLEU [49].

Readability indices are used to assign a grade level to text signifying its simplicity. All the readability indices are calculated using some combination of word weighting, syllable, letter, or word counts, and are shown to measure some level of simplicity. Automatic evaluation metrics, such as BLEU [49] and SARI [78], are widely used in TS research, with SARI [78] having specifically been developed for TS tasks. SARI is computed by comparing the generated simplifications with both the source and target references. It computes an average of F1-score for 3 n-gram overlap operations: additions, keeps, and deletions. Both BLEU [49] and SARI [78] are n-gram–based metrics, which may fail to capture the semantics of the generated text.

4.3 Methodology

Given a complex medical paragraph, the goal of this work is to generate a simplified paragraph that is concise and captures the salient information expressed in the complex text. To accomplish this, an RL-based simplification model is proposed, which optimizes multiple rewards during training, and is tuned using a paragraph-level medical TS data set.

Dataset

The Cochrane Database of Scientific Reviews is a health care database with information on a wide range of clinical topics. Each review includes a plain language summary (PLS) written by the authors who follow guidelines to structure the summaries. PLSs are supposed to be

clear, understandable, and accessible, especially for a general audience not familiar with the field of medicine. PLSs are highly heterogeneous in nature, and are not paired (i.e., for every complex sentence there may not be a corresponding simpler version). However, Devaraj et al. [18] used the Cochrane Database of Scientific Reviews data to produce a paired data set, which has 4459 pairs of complex-simple text, with each text containing less than 1024 tokens so that it can be fed into the BART [35] model for the purpose of TS. The pioneering data set developed by Devaraj et al. is used in this study for training the models and is publicly available [18].

TESLEA – Text Simplification Using Reinforcement Learning

The TS solution proposed for the task of simplifying complex medical text uses an RL-based simplification model, which optimizes multiple rewards (relevance reward, Flesch-Kincaid Grade rewards, and lexical simplicity rewards) to achieve a more complete and concise simplification. The following subsections introduce the computation of these rewards, along with the training procedure.

Relevance Reward

Relevance reward measures how well the semantics of the target text is captured in its simplified version. This is calculated by computing the cosine similarity between the target text embedding (E_T) and the generated text embedding (E_G). BioSentVec [13], a text embedding model trained on medical documents, is used to generate the text embeddings. The steps to calculate the relevance score are depicted in Algorithm 5.

The *RelevanceReward* function takes 3 arguments as input, namely, target text (T), generated text (G), and the embedding model (M). The function ComputeEmbedding

Algorithm 5: Relevance Reward

Input: T: Target text, G: Generated text, M: Embedding Model
Output: R_{cosine} : Relevance Reward
Variables: E_T : Target sentence embedding, E_G : Generated sentence embedding

1 **Function** `RelevanceReward(`T`,`G`,`M`)`
  `/* Compute sentence embedding for Target sentence. */`
2 $E_T \leftarrow$ ComputeEmbedding(T, M)
  `/* Compute sentence embedding for generated sentence. */`
3 $E_G \leftarrow$ ComputeEmbedding(G , M)
  `/* Compute Cosine Similarty. */`
4 $R_{cosine} \leftarrow \frac{E_T \cdot E_G}{\|E_T\| \cdot \|E_G\|}$
5 **return** R_{cosine}

takes the input text and embedding model (M) as input and generates the relevant text embedding. Finally, cosine similarity between generated text embedding (E_G) and target text embedding (E_T) is calculated to get the reward (Algorithm 5, line 4).

Flesch-Kincaid Grade Reward

FKGL refers to the grade level that must be attained to comprehend the presented information. A higher FKGL score indicates that the text is more complex, and a lower score indicates that the text is simpler. The FKGL for a text (S) is calculated using equation 4.1 [32]:

$$FKGL(S) = 0.38 \times \frac{\text{total words}}{\text{total sentences}} + 11.8 \times \frac{\text{total syllables}}{\text{total words}} - 15.59 \tag{4.1}$$

The FKGL reward (R_{Flesch}) is designed to reduce the complexity of generated text and is calculated as presented in Algorithm 6. In Algorithm 6, the function FleschKincaidReward takes 2 arguments as inputs, namely, generated text (G) and target text (T). The FKGLScore function calculates the FKGL for the given text. Once the FKGL for T and G is calculated, the Flesch-Kincaid reward (R_{Flesch}) is calculated as the relative difference

Algorithm 6: Flesch Kincaid Reward

Input: T: Target text, G: Generated text
Output: R_{flesch}: Flesch Kincaid Reward
Variables: r(T): Target text flesch kincaid grade level, r(G): generated text flesch kincaid grade level.

```
1 Function FleschKincaidReward(T,G)
2 |  r(T) ← FKGLScore(T) ;
3 |  r(G) ← FKGLScore(G) ;
4 |  R_flesch ← (r(T) − r(G))/r(T) ;
5 |  return R_flesch
```

between $r(T)$ and $r(G)$ (Algorithm 6, line 4), where $r(T)$ and $r(G)$ denote the FKGL of the target and generated text.

Lexical Simplicity Reward

Lexical simplicity is used to measure whether the words in the generated text (G) are simpler than the words in the source text (S). Laban et al. [34] proposed a lexical simplicity reward that uses the correlation between word difficulty and word frequency [8]. As word frequency follows zipf law, Laban et al. [34] used it to design the reward function, which involves calculating zipf frequency of newly inserted words, that is, $Z(G–S)$, and deleted words, that is, $Z(S–G)$. The lexical simplicity reward is defined in the same way as proposed by Laban et al. [34] and is described in Algorithm 7. The analysis of the data set proposed by Devaraj et al. [18] revealed that 87% of simple and complex pairs have a value of $\Delta Z(S, G) \approx 0.4$, where $\Delta Z(S, G) = Z(G–S)–Z(S–G)$ is the difference between the zipf frequency of inserted words and deleted words, with the value of lexical reward (Rlexical) scaled between 0 and 1.

In Algorithm 3, LexicalSimplicityReward requires the source text (S) and the generated text (G) as the inputs. Functions ZIPFInserted [7] and ZIPFDeleted [7] calculate the zipf frequency of newly inserted words and the deleted words. Finally, the lexical reward ($R_{lexical}$)

Algorithm 7: Lexical Simplicity Reward

Input: S: Source Text, G: Generated Text
Output: $R_{lexical}$: Lexical Simplicity Reward
Variables: $Z(G-S)$: Zipf frequncy of inserted words, $Z(S-G)$: Zipf frequncy of deleted words, $\Delta Z(S,G)$: Difference between Zipf frequency of inserted and Zipf frequency of deleted words

1 **Function** `LexicalSimplicityReward(`S`,`G`)`
  `/* Compute Zipf frequency of inserted words. */`
2 $Z(G-S) \leftarrow$ ZIPFInserted(G,S) ;
  `/* Compute Zipf frequency of deleted words. */`
3 $Z(S-G) \leftarrow$ ZIPFDeleted(G,S) ;
4 $\Delta Z(S,G) \leftarrow Z(G-S) - Z(S-G)$;
5 $R_{lexical} \leftarrow 1 - \frac{\Delta Z(S,G)-0.4}{0.4}$;
6 **return** $R_{lexical}$

is calculated and normalized, as described in line 5.

Training Procedure and Baseline Model

Pretrained BART

The baseline language model used in this study for performing simplification was BART [35], which is a transformer based encoder-decoder model that was pretrained using a denoising objective function. The decoder part of the model is autoregressive in nature, making it more suitable for sentence-generation tasks. Furthermore, the BART model achieves strong performance on natural language generation tasks such as summarization, which was demonstrated on XSum [43] and CNN/Daily Mail [42] data sets. In this case, a version of BART fine-tuned on XSUM [43] data set is being used.

Language Model Fine Tuning

Transformer based language models are pre-trained on large corpus of text and later finetuned on a downstream task by minimizing the maximum likelihoood loss (Lml) function. Consider a paired dataset C, where each instance consists of a source sentence containing n tokens $x = \{x_1, \cdots, x_n\}$ and target sequence containing m tokens $y = \{y_1, \cdots, y_m\}$, then the maximum likelihood loss (Lml) function is given in Equation 2 with the computation described in Algorithm 8.

$$Lml = -\sum_{t=1}^{m} logp_\theta(y_t|y_{<t}, x) \tag{4.2}$$

where θ are the model parameters and $y_{<t}$ denotes preceding tokens before the position t [53].

Algorithm 8: MLE Update

Input: D: Dictionary, θ: Language Model
Output: Lml: Maximum Likelihood Loss
Variables: *logits*: Output of the model
1 **Function** `MLEUpdate(`θ`,`D`)`
`/* FORWARD function returns the output of the model. */`
2 logits ← FORWARD(θ, D) ;
`/* Calculating maximum likelihood loss using` *logits* `and` D `*/`
3 Lml ← MLELoss(logits, D) ;
4 **return** Lml;

However, the results obtained by minimizing Lml are not always the best. There are two main reasons for degradation of results, the first is called "exposure bias" [59], which is when the model expects gold standard data at each step of training, but it does not receive such supervision during testing, which can result in accumulating errors during prediction. The second is called as "representation collapse" [3] which is degradation of pre-trained language model representations during fine-tuning. Ranzato et al. [59] avoided the problem of exposure bias by directly optimizing the specific discrete metric instead of minimzing the

maximum likelihood loss(Lml) via the help of reinforcement learning based algorithm called REINFORCE [75]. We use a variant of REINFORCE [75] called as "Self Critical Sequence Training(SCST)"[61] to directly optimize certain rewards specifically designed for TS, more information is provided in the following section.

Self-critical Sequence Training

Text simplification can be formulated as an reinforcement learning problem, where the "agent"(Language Model) interacts with the environment to take "action"(next word prediction) based on a learned "policy"(p_θ) defined by model parameters θ while observing some rewards (R). In this work, we use BART [35] as the language model. To learn an optimal policy that maximizes rewards, the REINFORCE [75] algorithm is used. Specifically we use REINFORCE with a baseline to stabilise the training procedure by using objective function Lpg with a baseline reward b given by:

$$Lpg = -(r(y^s) - b) \sum_{i=1}^{n} logp_\theta(y_i^s | y_1^s, ..., y_{i-1}^s, S) \tag{4.3}$$

where $p_\theta(y_i^s|..)$ denotes the probability of $i\text{-}th$ word conditioned on a previously generated sampled sequence by the model; $r(y^s)$ denotes the reward computed for a sentence generated using sampling; S denotes the source sentence and n is the length of the generated sentence. Rewards are computed as weighted sum of Relevance Reward (R_{cosine}), Flesch Kincaid Grade Reward (R_{flesch}) and Lexical Simplicity Reward ($R_{lexical}$) is given in Equation 4.4 and shown in Figure 4.2:

$$r(y^s) = \alpha \cdot R_{cosine} + \beta \cdot R_{flesch} + \delta \cdot R_{lexical} \tag{4.4}$$

where α, β, δ are the weights associated with the respective rewards. To approximate the baseline reward (b), Self Critical Sequence Training Strategy [61] is used. The baseline reward

(b) is calculated by computing reward values for a sentence which has been generated using greedy decoding $r(y^*)$ by the current model.

Algorithm 9: Self Critical Update

Input: D: Dictionary, M: Language Model
Output: Lpg: Policy Gradient Loss
Variables: y^s: Sampled Sentence, y^*: Greedy Sentence , n: length of generated sequence, $r(y^*)$: Reward for greedy sentence, $r(y^s)$: Reward for sampled sentence
Function `SelfCriticalUpdate(`θ`,`D`)`

```
/* Generate sentence using multinomial Sampling.                    */
```
$y^s \leftarrow$ GenerateSampleSentence(M, D) ;
```
/* Generate sentence using Greedy Decoding.                         */
```
$y^* \leftarrow$ GenerateGreedySentence(M, D) ;
```
/* Compute reward for greedy sentence.                              */
```
$r(y^*) \leftarrow$ ComputeRewards(y^*, D) ;
```
/* Compute reward for sampled sentence.                             */
```
$r(y^s) \leftarrow$ ComputeRewards(y^s, D) ;
$Lpg = -(r(y^s) - r(y^*)) \sum_{i=1}^{n} logp_\theta(y_i^s | y_1^s, ..., y_{i-1}^s)$;
return Lpg;

The loss function is defined in Equation 4.5 [61] and the way it is computed is described in Algorithm 9:

$$Lpg = -(r(y^s) - r(y^*)) \sum_{i=1}^{n} logp_\theta(y_i^s | y_1^s, \cdots, y_{i-1}^s, S) \tag{4.5}$$

where y^* denotes the sentence generated using greedy decoding.

Intuitively, by minimizing the loss described in equation 4.5, the likelihood of choosing the samples sequence (y^s) is promoted if the reward obtained for sampled sequence, $r(y^s)$, is greater than the reward obtained for the baseline rewards (i.e., the samples that return higher reward than $r(y^*)$). The samples that obtain a lower reward are subsequently suppressed. The model is trained using a combination of Lml and policy gradient loss similar to [50].

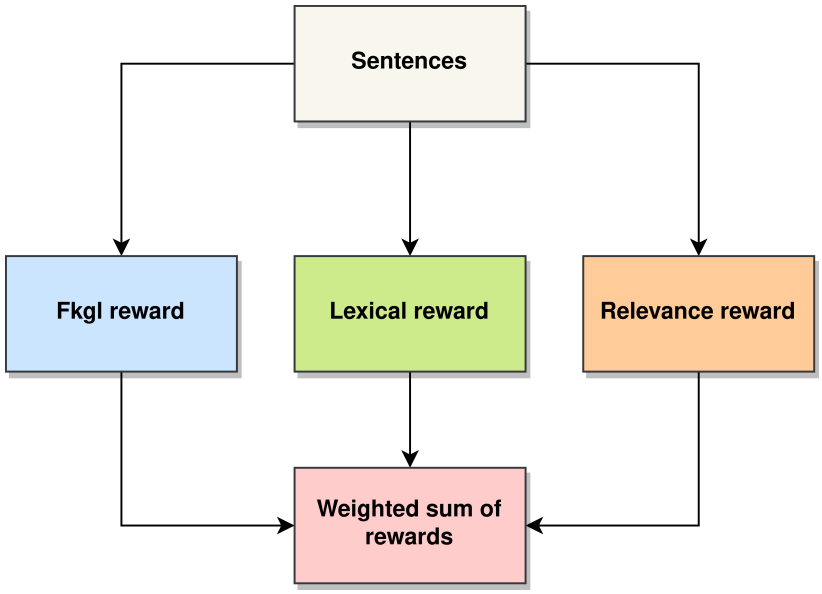

Figure 4.2: Compute Rewards function calculates a weighted sum of three rewards

The model is trained using a combination of maximum likelihood loss and policy gradient loss similar to [50]. The overall loss is given by:

$$L = \gamma Lpg + (1 - \gamma)Lml \tag{4.6}$$

where γ is a scaling factor that can finetuned.

Summary of Training Process

Overall, the training procedure follows a 2-step approach. As the pretrained BART [35] was not trained on the medical domain–related text, it was first fine-tuned on the document-level paired data set [18] by minimizing the Lml (maximum likelihood estimation (MLE) 4.2). In the second part, the fine-tuned BART model was trained further using RL. The RL procedure of TESLEA involves 2 steps: (1) the RL step and (2) the MLE optimization step, which are both shown in Figure 4.3 and further described in Algorithm 6. The given

simple-complex text pairs are converted to tokens as required by the BART model. In the MLE step, these tokens are used to compute logits from the model, and then finally MLE loss is computed. In the RL step, the model generates simplified text using 2 decoding strategies: (1) greedy decoding and (2) multinomial sampling. Rewards are computed as weighted sums (Figure 4.3) for sentences generated using both decoding strategies. These rewards are then used to calculate the loss for the RL step. Finally, a weighted sum of losses is computed that is used to estimate the gradients and update model parameters. All the hyperparameter settings used are included in Appendix B.

Algorithm 10: Training of Simplification System

Input: D_{pair} : Paired Dataset, N: Iterations, γ : weight, M: Language Model, M_f: Finetuned Language Model on paired Dataset(D_{pair})

Output: M: Language Model

$M \leftarrow M_f$
for $i = 1$ *to* N **do**
  for $batch \in D_{pair}$ **do**
    $D \leftarrow$ TOKENIZE(batch) ;
    `/* Calculate maximum likelihood loss. */`
    $Lml \leftarrow$ MLEUpdate(M, D) ;
    `/* Calculate policy gradient loss. */`
    $Lpg \leftarrow$ SelfCriticalUpdate(M,D) ;
    `/* Weighted sum of losses. */`
    $L = \gamma \cdot Lpg + (1 - \gamma) \cdot Lml$;
    Update model pararmeters with L ;
  end
end
return Language Model θ

Automatic Metrics

Two readability indices were used to perform automatic evaluations of the generated text, namely, FKGL and Automatic Readability Indices (ARIs). The SARI score is a standard

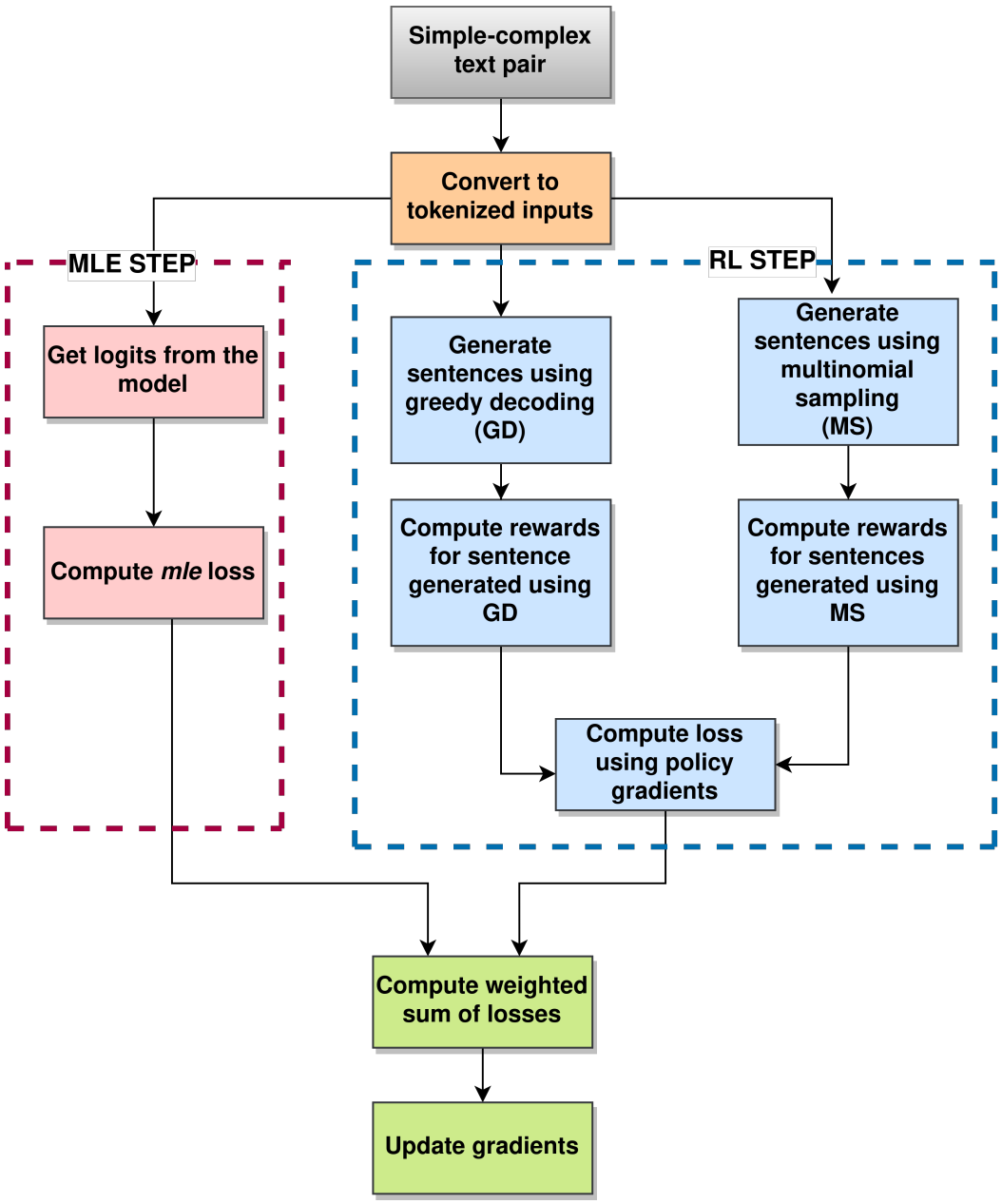

Figure 4.3: Reinforcement learning–based training procedure for TESLEA

metric for TS. The F-1 versions of ROUGE-1 and ROUGE-2 [37] scores were also reported. To measure the quality of the generated text, the criteria proposed by Yuan, Neubig, and Liu [80] were used, which are mentioned in the "Automatic Evaluation Metrics" section in Appendix-B. The criteria proposed by Yuan, Neubig, and Liu [80] can be automatically computed using a language model–based metric called "BARTScore." Further details on how to use BARTScore to measure the quality of the generated text are also mentioned in Appendix-B

Human Evaluations

In this study, 3-domain experts judge the quality of the generated text based on the factors mentioned in the previous section. The evaluators rate the text on a Likert scale from 1 to 5. First, simplified test data were generated using TESLEA, and then 51 generated paragraphs were randomly selected, creating 3 subsets containing 17 paragraphs each. Every evaluator was presented with 2 subsets, that is, a total of 34 complex-simple TESLEA-generated paragraphs. The evaluations were conducted via Google Forms, and the human annotators were asked to measure the quality of simplification for informativeness (INFO), fluency (FLU), coherence (COH), factuality (FAC), and adequacy (ADE) (Figure 4.4). All the data collected were stored in CSV files for statistical analysis.

4.4 Results

This section consists of 3 subsections, namely,

- Baseline Models
- Automatic Evaluations

COMPLEX MEDICAL PARAGRAPH

A total of 38 studies involving 7843 children were included. Following educational intervention delivered to children, their parents or both, there was a significantly reduced risk of subsequent emergency department visits (RR 0.73, 95% CI 0.65 to 0.81, N = 3008) and hospital admissions (RR 0.79, 95% CI 0.69 to 0.92, N = 4019) compared with control. There were also fewer unscheduled doctor visits (RR 0.68, 95% CI 0.57 to 0.81, N = 1009). Very few data were available for other outcomes (FEV1, PEF, rescue medication use, quality of life or symptoms) and there was no statistically significant difference between education and control. Asthma education aimed at children and their carers who present to the emergency department for acute exacerbations can result in lower risk of future emergency department presentation and hospital admission. There remains uncertainty as to the long-term effect of education on other markers of asthma morbidity such as quality of life, symptoms and lung function. It remains unclear as to what type, duration and intensity of educational packages are the most effective in reducing acute care utilisation.

GENERATED SIMPLE MEDICAL PARAGRAPH

This review of studies found that education aimed at children and their carers reduces the need for future emergency department visits for acute exacerbations in children aged four to 16 years who suffer an asthma attack. Although education programmes have been effective at reducing the emergency department visit, there is uncertainty as to whether education programmes can have a long-term impact on other markers of asthma morbidity, such as quality of life, symptoms and breathing patterns.

QUESTIONS

- Rate the Generated text on a scale to 1 to 5 considering the Informativeness
  1. No relevant information is retained in generated text
  2. Partial relevant information is retained in generated text
  3. Neutral/ Undecided
  4. Significant relevant information is retained in generated text
  5. All relevant information is retained in generated text
- Rate the Generated text on a scale to 1 to 5 considering the Fluency
  1. Fluency is lost in the generated text
  2. Fluency is partially lost in the generated text.
  3. Neutral/ Undecided
  4. Fluency is partially maintained in the generated text.
  5. Fluency is maintained in the generated text.
- Rate the Generated text on a scale to 1 to 5 considering the Coherence
  1. Coherence is lost in the generated text.
  2. Coherence is partially lost in the generated text.
  3. Neutral/ Undecided.
  4. Coherence is partially maintained in the generated text.
  5. Coherence is maintained in the generated text.
- Rate the Generated text on a scale to 1 to 5 considering the Factuality
  1. Factuality is lost in the generated text
  2. Factuality is partially lost in the generated text.
  3. Neutral/ Undecided
  4. Factuality is partially maintained in the generated text.
  5. Factuality is maintained in the generated text.
- Rate the Generated text on a scale to 1 to 5 considering the Adequacy.
  1. Adequacy is lost in the generated text
  2. Adequacy is partially lost in the generated text
  3. Neutral/ Undecided
  4. Adequacy is partially maintained in the generated text
  5. Adequacy is maintained in the generated text.

Figure 4.4: A sample question seen by the human annotator.

- Human Evaluations.

The first section highlights the baseline models used for comparison and analysis. The second section discusses the results obtained by performing automatic evaluations of the model. The third and final section discusses the results obtained from human assessments and analyzes the relationship between human annotations and automatic metrics.

Baseline Models

TESLEA is compared with other strong baseline models and their details are discussed below:

- **BART-Fine-tuned**: BART-Fine-tuned is a BART-large model fine-tuned using a maximum likelihood loss on the data set proposed by Devaraj et al. [18]. Studies have shown that large pretrained models often perform competitively when fine-tuned for downstream tasks, thus making this a strong competitor.

- **BART-UL**: Devaraj et al. [18] also proposed BART-UL for paragraph-level medical TS. It is the first model to perform paragraph-level medical TS and has achieved strong results on automated metrics. BART-UL was trained using an unlikelihood objective function that penalizes the model for generating technical words (i.e., complex words). Further details on the training procedure of BART-UL are described in Appendix-A

- **MUSS**: MUSS [41] is a BART-based language model that was trained by mining paraphrases from the CCNet corpus [74]. MUSS was trained on a data set consisting of 1 million paraphrases, helping it achieve a strong SARI score. Although MUSS is trained on a sentence-level data set, it still serves as a strong baseline for comparison. Further details on the training procedure for MUSS are discussed in Appendix-A

- **Keep it Simple (KIS)**: Laban et al. [34] proposed an unsupervised approach for paragraph-level TS. KIS is trained using RL and uses the GPT-2 model as a backbone. KIS has shown strong performance on SARI scores beating many supervised and unsupervised TS approaches. Additional details on the training procedure for KIS are described in Appendix-A

- **PEGASUS models**: PEGASUS is a transformer-based encoder-decoder model that has achieved state-of-the-art results on many text-summarization data sets. It was specifically designed for the task of text summarization. In our analysis, we used two variants of PEGASUS models, namely,

  - PEGASUS-large (PL), the large variant of Pegasus model
  - PEGASUS-pubmed-large (PPL), the large variant of the PEGASUS model that was pretrained on a PubMed data set.

  Both the PEGASUS models were fine-tuned using maximum likelihood loss on the data set proposed by Devaraj et al. [18]. For more information regarding the PEGASUS model, the readers are suggested to refer to [82].

Results of Automatic Metrics

The metrics used for automatic evaluation are FKGL, ARI, ROUGE-1, ROUGE-2, SARI, and BARTScore. The mean readability indices scores (i.e., FKGL and ARI) obtained by various models are reported in Table 4.1. ROUGE-1, ROUGE-2, and SARI scores are reported in Table 4.2 and BARTScore is reported in Table 4.3.

Readability Indices, ROUGE, and SARI Scores

The readability indices scores reported in Table 4.1 suggest that the FKGL scores obtained by TESLEA are better (i.e., a lower score) when compared with the FKGL scores obtained by comparing technical abstracts (i.e., complex medical paragraphs available in the data set) with the gold-standard references (i.e., simple medical paragraphs corresponding to the complex medical paragraphs). Moreover, TESLEA achieves the lowest FKGL score (11.84) when compared with baseline models, indicating significant improvement in the TS. The results suggest that

- BART-based transformer models are capable of performing simplification at the paragraph level such that the outputs are at a reduced reading level (FKGL) when compared with technical abstracts, gold-standard references, and baseline models.
- The proposed method to optimize TS-specific rewards allows the generation of text with greater readability than even the gold-standard references, as indicated by the FKGL scores in Table 4.1.
- The reduction in FKGL scores can be explained by the fact that FKGL was a part of a reward (R_{Flesch}) that was directly being optimized.

In addition, we report the SARI [78] and ROUGE scores [37] as shown in Table 4.2. SARI is a standard automatic metric used in sentence-level TS tasks. The ROUGE score is another standard metric in text summarization tasks. The findings indicate that TESLEA is capable of performing at the same level as the baseline models on ROUGE scores. Furthermore, TESLEA has been observed to attain a similar level of performance as BART-UL on SARI score, which suggests that the generated outputs by TESLEA are simplified and consistent with the previous baselines. Although the models are achieveing the same SARI scores, there

		FKGL	ARI
Baseline Text	**Technical Abstracts**	14.42	15.58
	Gold References	13.11	15.08
Model Generated	**BART-Finetuned**	13.45	15.32
	BART-UL	11.97	**13.73**
	TESLEA	**11.84**	13.82
	MUSS	14.29	17.29
	KIS	14.15	17.05
	PL	14.53	17.55
	PPL	16.35	19.8

Table 4.1: Flesch Kincaid Grade Level (FKGL), Automatic Readability Index (ARI) for the generated text. TESLEA significantly reduces FKGL and ARI scores when compared to plain language summaries. Bold indicates best scores.

are differences in the quality of text generated by these models and these are explained in the subsequent subsection subsection.

Model	ROUGE-1	ROUGE-2	SARI
BART-Finetuned	0.40	0.11	0.39
BART-UL	0.38	0.14	**0.40**
TESLEA	0.39	0.11	**0.40**
MUSS	0.23	0.03	0.34
KIS	0.23	0.03	0.32
PL	**0.44**	**0.18**	**0.40**
PPL	0.42	0.16	**0.40**

Table 4.2: ROUGE-1, ROUGE-2 and SARI scores for the generated text. TESLEA achieves similar performance to other models. Higher scores of ROUGE-1, ROUGE-2, and SARI are desirable.

Text Quality Measure

There has been significant progress in designing automatic metrics that are able to capture linguistic quality of the text generated by language models. One such metric that is able

to measure the quality of generated text is BARTScore [80]. BARTScore has shown strong correlation with human assessments on various tasks ranging from machine translation to text summarization. BARTScore has 4 different metrics (i.e., Faithfulness Score, Precision, Recall, F-score), which can be used to measure different qualities of generated text. Further details on how to use BARTScore are mentioned in Appendix B. According to the analysis

Models	**Faithfulness-Score**	**F-Score**
BART-Finetuned	0.137	0.078
BART-UL	0.242	0.061
TESLEA	**0.366**	**0.097**
MUSS	0.031	0.029
KIS	0.030	0.028
PL	0.197	0.073
PPL	0.29	0.063

Table 4.3: Faithfulness-Score and F-score for the generated text by the models. TESLEA achieves the highest faithfulness score and F-score. Higher scores of Faithfulness and F-score are desirable.

conducted by Yuan, Neubig, and Liu [80], Faithfulness Score measures 3 aspects of generated text via COH, FLU, and FAC. The F-score measures 2 aspects of generated text (INFO and ADE). In our analysis, we use these 2 variants of BARTScore to measure COH, FLU, FAC, INFO, and ADE. TESLEA achieves the highest values (Table 4.3) of Faithfulness Score (0.366) and F-score (0.097), indicating that the rewards designed for the purpose of TS not only help the model in generating simplified text but also on some level preserve the quality of generated text. The F-scores of all the models are relatively poor (i.e., scores closer to 1 are desirable). One of the reasons for low F-scores could be the introduction of misinformation or hallucinations in the generated text, a common problem for language models, which could be addressed by adapting training strategies that focus on INFO via the help of rewards or objective functions. For qualitative analysis we randomly selected 50 sentences from the test

data and calculated the average number of tokens based on BART model vocabulary. For the readability measure, we calculated the FKGL scores of these generated texts and noted any textual inconsistencies such as misinformation. The analysis revealed that the text generated by most models was significantly smaller than the gold-standard references (Table 4.4). Furthermore, TESLEA- and BART-UL–generated texts were significantly shorter compared with other baseline models and TESLEA had the lowest FKGL score among all the models as depicted in Table 4.4. From a qualitative point of view, the sentences generated by most

Model	**Number of Tokens**	**FKGL**
Technical Abstracts	498.11	14.37
Gold References	269.74	12.77
BART-Finetuned	143.70	12.58
TESLEA	**131.37**	**12.34**
BART-UL	145.08	12.66
KIS	187.59	13.78
MUSS	193.07	13.86
PL	272.04	13.93
PPL	150.00	15.09

Table 4.4: Average Number of tokens and Average FKGL scores for selected samples.

baseline models involve significant duplication of text from the original complex medical paragraph. The outputs generated by the KIS model were incomplete and appear "noisy" in nature. One of the reasons for the noise generation could be because of unstable training due to lack of a huge corpus of domain-specific data. BART-UL–generated paragraphs are simplified as indicated by the FKGL and ARI scores, but they are extractive in nature (ie, the model learns to select simplified sentences from the original medical paragraph and combines them to form a simplification). PEGASUS-pubmed- large–generated paragraphs are also extractive in nature and similar to BART-UL–generated paragraphs, but it was observed that they were grammatically inconsistent. In contrast to baseline models, the text

generated by TESLEA was concise, semantically relevant, and simple, without involving any medical domain–related complex vocabulary. Figures 4.5 and 4.6 shows an example of text generated by all the models, with blue text indicating the copied text. In addition to the duplicated text, the models also induced misinformation in the generated text. The most common form of induced misinformation observed was "The evidence is current up to [date]," as shown in Figure 4.7. This text error occurred due to the structure of the data (i.e., PLS contains statements related to this research, but these statements were not in the original text; thus, the model attempted to add these statements to the generated text although it is not factually correct). Thus considerable attention should be paid to including FAC measures in the training regime of these models. For a more complete assessment of the quality of simplification, human evaluation was conducted using domain experts for the text generated by TESLEA.

COMPLEX MEDICAL PARAGRAPH

A total of 38 studies involving 7843 children were included. Following educational intervention delivered to children, their parents, or both, there was a significantly reduced risk of subsequent emergency department visits(RR 0.73, 95% CI 0.65 to 0.81, N = 3008)and hospital admissions (RR 0.79, 95% CI 0.69 to 0.92, N = 4019) compared with control. There were also fewer unscheduled doctor visits(RR 0.68, 95% CI 0.57 to 0.81, N = 1009). Very few data were available for other outcomes (FEV1, PEF, rescue medication use, quality of life or symptoms), and there was no statistically significant difference between education and control. Asthma education aimed at children and their carers who present to the emergency department for acute exacerbations can result in a lower risk of future emergency department presentation and hospital admission. There remains uncertainty as to the long-term effect of education on other markers of asthma morbidity, such as quality of life, symptoms, and lung function. It remains unclear as to what type, duration, and intensity of educational packages are the most effective in reducing acute care utilization.

BART FINETUNED GENERATED PARAGRAPH

A total of 38 studies involving 7843 children were included in this review. Following educational intervention delivered to children, their parents, or both, there was a significantly reduced risk of subsequent emergency department visits (RR 0.73, 95% CI 0.65 to 0.81, N = 3008) and hospital admissions compared with control. There were also fewer unscheduled doctor visits. There remains uncertainty as to the long-term effect of education on other markers of asthma morbidity, such as quality of life, symptoms, and lung function.

PPL GENERATED PARAGRAPH

We review found that education aimed at children and their carers who present to the emergency department for acute exacerbations can result in a lower risk of future emergency department visits and hospital admission. We remains uncertain as to the long-term effect of education on other markers of asthma morbidity such as quality of life, symptoms and lung function.

KIS GENERATED PARAGRAPH

A total of 38 studies involving 7843 children were included. Following educational intervention, parents or both, there was a significant reduction in the risk of emergency department visits for children who had experienced trouble at school. RR 0.73, the most common estimate of the risk of emergency department visits, was also found to be associated with asthma symptoms (RR 0.65, 95% CI 0.65 to 0.81, N = 3008).

Figure 4.5: Comparison of Text Generated by all the models.

MUSS GENERATED PARAGRAPH

A total of 38 studies involving 7843 children were included. Following educational intervention delivered to children, their parents or both, there was a significantly reduced risk of subsequent emergency department visits (RR 0.73, 95% CI 0.65 to 0.81, N = 3008) and hospital admissions (RR 0.79, 95% CI 0.69 to 0.92, N = 4019). There were also fewer unscheduled doctor visits (RR 0.68, 95% CI 0.57 to 0.81, N = 1009). There was no statistically significant difference between education and control. Asthma education aimed at children and their carers who present to the emergency department for acute exacerbations can result in lower risk of future emergency department presentation and hospital admission. It remains unclear as to what type, duration and intensity of educational packages are the most effective in reducing acute care utilisation.

PL GENERATED PARAGRAPH

This review of 38 studies involving 7843 children found that asthma education delivered to children, their parents or both, can reduce the risk of future emergency department visits and hospital admissions. There were also fewer unscheduled doctor visits. Very few data were available for other outcomes (FEV1, PEF, rescue medication use, quality of life or symptoms) and there was no statistically significant difference between education and control. Asthma education aimed at children and their carers who present to the emergency department for acute exacerbations can result in lower risk of future emergency department presentation and hospital admission. There remains uncertainty as to the long-term effect of education on other markers of asthma morbidity such as quality of life, symptoms and lung function. It remains unclear as to what type, duration and intensity of educational packages are the most effective in reducing acute care utilisation.

BART-UL GENERATED PARAGRAPH

This systematic review identified 38 studies involving 7843 children. These studies found that asthma education aimed at children and their carers who present to the emergency department for acute exacerbations can result in lower risk of future emergency department presentation and hospital admission. However, there is uncertainty as to what type, duration and intensity of educational packages are the most effective in reducing acute care utilisation. There remains uncertainty about the long-term effect of education on other markers of asthma morbidity such as quality of life, symptoms and lung function.

TESLEA GENERATED PARAGRAPH

This review of studies found that education aimed at children and their carers reduces the need for future emergency department visits for acute exacerbations in children who suffer from an asthma attack. Although education programmes have been effective at reducing the emergency department visits, there is uncertainty as to whether education programmes can have a long-term impact on other markers of asthma morbidity, such as quality of life, symptoms and breathing patterns.

Figure 4.6: Comparison of Text Generated by all the models.

GENERATED TEXT

Twelve trials including a total of 1211 trials were included in this review. Seven trials compared IVIg with PE and compared it with PE. The evidence is current up to July 2013. These trials were from all over the world and include people with CIDSL and MS and include people with and without MS from all walks of life. The findings of this review suggest that, in severe cases of MS, IVIg, given within two weeks of onset of the disease, hastens recovery as much as PE therapy.

Figure 4.7: Example of misinformation found in Generated text

Human Evaluations

For this research, 3 domain experts assessed the quality of generated text, based on factors of INFO, FLU, COH, FAC, and ADE, as proposed by Yuan, Neubig, and Liu [80], which are discussed in Appendix B. To measure inter-rater reliability, the percentage agreement between the annotators is calculated, and the results are shown in Table 4.5. The average percentage agreement for the factors of FLU, COH, FAC, and ADE is the highest, indicating that annotators agree among their evaluations. The average Likert score for each factor is also reported by each rater (Table 4.6). From the data mentioned in Table 4.6, the raters think that the COH and FLU have the highest quality, with the ADE, FAC, and INFO also rated reasonably high. To further assess whether results obtained by automated metrics truly signify an improvement in the quality of generated text by TESLEA, the Spearman rank correlation coefficient was calculated between human ratings and the automatic metrics for all 51 generated paragraphs (text), with the results shown in Table 4.7. The BARTScore has the highest correlation with human ratings for FLU, FAC, COH, and ADE compared with other metrics.

	INFO	FLU	FAC	COH	ADE
A1 and A2	82.35	82.35	82.35	70.59	82.35
A1 and A3	70.59	58.82	70.59	70.59	70.59
A3 and A2	52.94	70.59	74.51	74.51	64.71
Average (% agreement)	**68.63**	**70.59**	**74.51**	**74.51**	**72.55**

Table 4.5: Average percent inter-rater agreement where A1 stands for Annotator 1, A2 indicates Annotator 2 and A3 indicates Annotator 3.

	INFO	FLU	FAC	COH	ADE
A1	3.82	4.12	3.91	3.97	3.76
A2	3.50	4.97	3.59	4.82	3.68
A3	4.06	3.94	3.85	3.94	3.85
ALS	3.79	**4.34**	3.78	**4.24**	3.76

Table 4.6: Average Likert score by each rater for INFO, FLU, COH, ADE. ALS stands for average Likert score.

Metric	INFO	FLU	FAC	COH	ADE
ROUGE-1	0.18	-0.04	-0.01	-0.05	0.06
ROUGE-2	0.08	-0.01	-0.05	-0.04	0.05
SARI	0.09	-0.66	-0.13	-0.01	0.01
BARTScore	0.08	**0.32**	**0.38**	**0.22**	0.07

Table 4.7: Spearman's Rank correlation coefficient between automatic metrics and human ratings for text generated by TESLEA. Bold indicates the best result.

4.5 Discussion

Principal Findings

The most up-to-date research about biomedicine is often inaccessible to the general public due to the domain-specific medical terminology. A way to address this problem is by creating a system that converts complex medical information into a simpler form, thus making

it accessible to everyone. In this study, a TS approach was developed that can automatically simplify complex medical paragraphs while maintaining the quality of the generated text. The proposed approach trains the transformer-based BART model to optimize rewards specific for TS, resulting in increased simplicity. The BART model is trained using the proposed RL method to optimize certain rewards that help generate simpler text while maintaining the quality of generated text. As a result, the trained model generates simplified text that reduces the complexity of the original text by 2-grade points, when measured using the FKGL [32]. From the results obtained, it can be concluded that TESLEA is effective in generating simpler text compared with technical abstracts, the gold-standard references (i.e., simple medical paragraphs corresponding to complex medical paragraphs), and the baseline models. Although previous work [18] developed baseline models for this task, to the best of our knowledge, this is the first time RL is being applied to the field of medical TS. Moreover, previous studies failed to analyze the quality of the generated text, which this study measures via the factors of FLU, FAC, COH, ADE, and INFO. Manual evaluations of TESLEA-generated text were conducted with the help of domain experts using the aforesaid factors and further research was conducted to analyze which automatic metrics agree with manual annotations using the Spearman rank correlation coefficient. The analysis revealed that BARTScore [80] best correlates with the human annotations when evaluated for a text generated by TESLEA, indicating that TESLEA learns to generate semantically relevant and fluent text, which conveys the essential information mentioned in the complex medical paragraph. These results suggest that (1) TESLEA can perform TS of medical paragraphs such that outputs are simple and maintain the quality, (2) the rewards optimized by TESLEA help the model capture syntactic and semantic information, increasing the FLU and COH of outputs, as witnessed when the outputs are evaluated by BARTScore and human annotators.

Limitations and Future Work

Although this research is a significant contribution to the literature on medical TS, the proposed approach does have a few limitations, addressing which can result in even better outputs. TESLEA can generate simpler versions of the text, but in some instances, it induces misinformation, resulting in reduced FAC and INFO of the generated text. Therefore, there is a need to design rewards that consider the FAC and INFO of the generated text. We also plan to conduct extensive human evaluations on a large scale for the text generated by various models (eg, KIS, BART-UL) using domain experts (i.e., physicians and medical students). Transformer-based language models are sensitive to the pretraining regime, so a possible next step is to pretrain a language model on domain-specific raw data sets such as PubMed [65], which will help develop domain-specific vocabulary for the model. Including these strategies may help in increasing the simplicity of the generated text.

Conclusion

The interest in and need for TS in the medical domain are of growing interest as the quantity of data is continuously increasing. Automated systems, such as the one proposed in this paper, can dramatically increase accessibility to information for the general public. This work not only provides a technical solution for automated TS but also lays out and addresses the challenges of evaluating the outputs of such systems, which can be highly subjective. It is the author's sincere hope that this work allows other researchers to build on and improve the quality of similar efforts.

Chapter 5

Conclusion

In this thesis, at first, Chapter 2 offers a comprehensive and informative overview of the most commonly used datasets, methods, and evaluation techniques in the field of text simplification, providing valuable context and understanding for the reader. Chapter 3 presents a detailed bibliometric analysis of the TS literature collected from 2001 to 2022, addressing six research questions that shed light on the evolution of TS over the years, important researchers in the field, country-wise research output, the evolution of research collaborations over the years, and trends in the field of TS. This analysis guided us to apply text simplification to the field of medicine. In Chapter 4, a unique approach called TESLEA is proposed, which leverages a high-performing transformer-based language model and a reinforcement learning training procedure to perform medical text simplification. TESLEA can simplify paragraph-level medical text data and outperforms standard approaches in TS based on automatic metrics. TESLEA can simplify medical text data by more than 2-grade points when measured on the FKGL scale. Human evaluation of TESLEA's output is also conducted with the help of domain experts (i.e., medical health professionals) via various factors highlighted in Chapter 4. The research presented in this thesis aims to encourage further analysis

and development of approaches and applications of TS. Research articles from various fields often use complicated language, which can be hard for non-experts to understand. Moreover, these articles may be available in different languages like French, Spanish, German, etc. Simplifying such technical language manually can be costly and time-consuming. However, with the advent of Large Language Models (LLMs) like InstructGPT [46], GPT3 [9], etc, there is a strong potential for building domain-specific multilingual text simplification solutions. By leveraging LLMs, we can develop scalable and efficient Automatic Text Simplification (ATS) systems. These systems are especially crucial in fields like medicine and finance, where understanding research is essential for making informed decisions. To build better ATS systems, developing large text simplification corpora would also be helpful. By combining this corpus with LLMs, can lead to building more accurate and comprehensive models for simplifying technical language.

Bibliography

[1] Vıctor M Garro Abarca, Pedro R Palos-Sanchez, and Enrique Rus-Arias. "Working in virtual teams: a systematic literature review and a bibliometric analysis". In: *IEEE access* 8 (2020), pp. 168923–168940.

[2] Muhammad Afzal et al. "Clinical context–aware biomedical text summarization using deep neural network: model development and validation". In: *Journal of medical Internet research* 22.10 (2020), e19810.

[3] Armen Aghajanyan et al. "Better Fine-Tuning by Reducing Representational Collapse". In: *International Conference on Learning Representations*. 2020.

[4] Fernando Alva-Manchego et al. "ASSET: A Dataset for Tuning and Evaluation of Sentence Simplification Models with Multiple Rewriting Transformations". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, July 2020, pp. 4668–4679. DOI: 10.18653/v1/2020.acl-main.424. URL: https://aclanthology.org/2020.acl-main.424.

[5] Regina Barzilay and Noemie Elhadad. "Sentence Alignment for Monolingual Comparable Corpora". In: *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing*. 2003, pp. 25–32. URL: https://aclanthology.org/W03-1004.

[6] Joachim Bingel, Gustavo Paetzold, and Anders Søgaard. "Lexi: A tool for adaptive, personalized text simplification". In: *Proceedings of the 27th International Conference on Computational Linguistics*. 2018, pp. 245–258.

[7] Johannes Bjerva et al. "The Meaning Factory: Formal Semantics for Recognizing Textual Entailment and Determining Semantic Similarity." In: *SemEval@ COLING*. 2014, pp. 642–646.

[8] Hunter M Breland. "Word frequency and word difficulty: A comparison of counts in four corpora". In: *Psychological Science* 7.2 (1996), pp. 96–99.

[9] Tom Brown et al. "Language models are few-shot learners". In: *Advances in neural information processing systems* 33 (2020), pp. 1877–1901.

[10] John A Carroll et al. "Simplifying text for language-impaired readers". In: *Ninth Conference of the European Chapter of the Association for Computational Linguistics*. 1999, pp. 269–270.

[11] Dhivya Chandrasekaran and Vijay Mago. "Comparative analysis of word embeddings in assessing semantic similarity of complex sentences". In: *IEEE Access* 9 (2021), pp. 166395–166408.

[12] Dhivya Chandrasekaran and Vijay Mago. "Evolution of Semantic Similarity—A Survey". In: *ACM Computing Surveys (CSUR)* 54.2 (2021), pp. 1–37.

[13] Qingyu Chen, Yifan Peng, and Zhiyong Lu. "BioSentVec: creating sentence embeddings for biomedical texts". In: *2019 IEEE International Conference on Healthcare Informatics (ICHI)*. IEEE. 2019, pp. 1–5.

[14] Xieling Chen et al. "A bibliometric analysis of natural language processing in medical research". In: *BMC medical informatics and decision making* 18.1 (2018), pp. 1–14.

[15] William Coster and David Kauchak. "Simple English Wikipedia: a new text simplification task". In: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. 2011, pp. 665–669.

[16] Jan De Belder and Marie-Francine Moens. "Text simplification for children". In: *Prroceedings of the SIGIR workshop on accessible search systems*. ACM; New York. 2010, pp. 19–26.

[17] Ashwin Devaraj et al. *Paragraph level medical Text Simplification Dataset.* `https://github.com/AshOlogn/Paragraph-level-Simplification-of-Medical-Texts`. 2021.

[18] Ashwin Devaraj et al. "Paragraph-level Simplification of Medical Texts". In: *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 2021, pp. 4972–4984.

[19] Naveen Donthu et al. "How to conduct a bibliometric analysis: An overview and guidelines". In: *Journal of Business Research* 133 (2021), pp. 285–296.

[20] Richard Evans, Constantin Orasan, and Iustin Dornescu. "An evaluation of syntactic simplification rules for people with autism". In: Association for Computational Linguistics. 2014.

[21] Jesse Fagan et al. "Assessing research collaboration through co-authorship network analysis". In: *The journal of research administration* 49.1 (2018), p. 76.

[22] Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. "PPDB: The paraphrase database". In: *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 2013, pp. 758–764.

[23] Dmitri Goldenberg. "Social network analysis: From graph theory to applications with python". In: *arXiv preprint arXiv:2102.10014* (2021).

[24] Sujatha Das Gollapalli and Xiaoli Li. "EMNLP versus ACL: Analyzing NLP research over time". In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. 2015, pp. 2002–2006.

[25] Annette Rios Gonzales et al. "A new dataset and efficient baselines for document-level text simplification in German". In: *Proceedings of the Third Workshop on New Frontiers in Summarization*. 2021, pp. 152–161.

[26] Maarten Grootendorst. "BERTopic: Neural topic modeling with a class-based TF-IDF procedure". In: *arXiv preprint arXiv:2203.05794* (2022).

[27] Anne Wil Harzing. *Publish or perish*. Feb. 2016. URL: `https://harzing.com/resources/publish-or-perish`.

[28] Chao Jiang et al. "Neural CRF Model for Sentence Alignment in Text Simplification". In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 2020, pp. 7943–7960.

[29] Mandar Joshi et al. "SpanBERT: Improving Pre-training by Representing and Predicting Spans". In: *Transactions of the Association for Computational Linguistics* 8 (2020), pp. 64–77.

[30] Tomoyuki Kajiwara and Mamoru Komachi. "Complex word identification based on frequency in a learner corpus". In: *Proceedings of the thirteenth workshop on innovative use of NLP for building educational applications*. 2018, pp. 195–199.

[31] Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: *Proceedings of NAACL-HLT*. 2019, pp. 4171–4186.

[32] J Peter Kincaid et al. *Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for navy enlisted personnel*. Tech. rep. Naval Technical Training Command Millington TN Research Branch, 1975.

[33] David Klaper, Sarah Ebling, and Martin Volk. "Building a German/Simple German Parallel Corpus for Automatic Text Simplification". In: *ACL 2013* (2013), p. 11.

[34] Philippe Laban et al. "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)*. 2021, pp. 6365–6378.

[35] Mike Lewis et al. "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*. 2020, pp. 7871–7880.

[36] Yang Li et al. "A bibliometric analysis on deep learning during 2007–2019". In: *International Journal of Machine Learning and Cybernetics* 11.12 (2020), pp. 2807–2826.

[37] Chin-Yew Lin. "Rouge: A package for automatic evaluation of summaries". In: *Text summarization branches out*. 2004, pp. 74–81.

[38] Jun Liu and Yuji Matsumoto. "Simplification of example sentences for learners of japanese functional expressions". In: *Proceedings of the 3rd Workshop on Natural Language Processing Techniques for Educational Applications (NLPTEA2016)*. 2016, pp. 1–5.

[39] Lin Liu et al. "An overview of topic modeling and its current applications in bioinformatics". In: *SpringerPlus* 5.1 (2016), pp. 1–22.

[40] Louis Martin et al. "Controllable Sentence Simplification". In: *Proceedings of the 12th Language Resources and Evaluation Conference*. 2020, pp. 4689–4698.

[41] Louis Martin et al. "MUSS: multilingual unsupervised sentence simplification by mining paraphrases". In: *arXiv preprint arXiv:2005.00352* (2020).

[42] Ramesh Nallapati et al. "Abstractive Text Summarization using Sequence-to-sequence RNNs and Beyond". In: *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*. 2016, pp. 280–290.

[43] Shashi Narayan, Shay B Cohen, and Mirella Lapata. "Don't Give Me the Details, Just the Summary! Topic-Aware Convolutional Neural Networks for Extreme Summarization". In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. 2018, pp. 1797–1807.

[44] Rani Nelken and Stuart M. Shieber. "Towards Robust Context-Sensitive Sentence Alignment for Monolingual Corpora". In: *11th Conference of the European Chapter of the Association for Computational Linguistics*. Trento, Italy: Association for Computational Linguistics, Apr. 2006, pp. 1611–168. URL: https://aclanthology.org/E06-1021.

[45] Sergiu Nisioi et al. "Exploring neural text simplification models". In: *Proceedings of the 55th annual meeting of the association for computational linguistics (volume 2: Short papers)*. 2017, pp. 85–91.

[46] Long Ouyang et al. "Training language models to follow instructions with human feedback". In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 27730–27744.

[47] Halil Ziya Özcan and Zekerya Batur. "A Bibliometric Analysis of Articles on Text Simplification: Sample of Scopus Database". In: *International Journal of Education and Literacy Studies* 9.2 (2021), pp. 24–40.

[48] Gustavo Paetzold and Lucia Specia. "Unsupervised lexical simplification for non-native speakers". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 30. 1. 2016.

[49] Kishore Papineni et al. "Bleu: a method for automatic evaluation of machine translation". In: *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*. 2002, pp. 311–318.

[50] Romain Paulus, Caiming Xiong, and Richard Socher. "A Deep Reinforced Model for Abstractive Summarization". In: *International Conference on Learning Representations*. 2018.

[51] Atish Pawar and Vijay Mago. "Challenging the boundaries of unsupervised learning for semantic similarity". In: *IEEE Access* 7 (2019), pp. 16291–16308.

[52] Atharva Phatak et al. "Medical Text Simplification Using Reinforcement Learning (TESLEA): Deep Learning–Based Text Simplification Approach". In: *JMIR Med Inform* 10.11 (Nov. 2022), e38095. ISSN: 2291-9694. DOI: 10.2196/38095. URL: http://www.ncbi.nlm.nih.gov/pubmed/36399375.

[53] Weizhen Qi et al. "ProphetNet: Predicting Future N-gram for Sequence-to-SequencePre-training". In: *Findings of the Association for Computational Linguistics: EMNLP 2020*. 2020, pp. 2401–2410.

[54] Jipeng Qiang et al. "Lexical simplification with pretrained encoders". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 34. 05. 2020, pp. 8649–8656.

[55] Mohiuddin Qudar and Vijay Mago. "A survey on language models". In: *Association for Computing Machinery* 1 (2020).

[56] Dragomir R Radev et al. "A bibliometric and network analysis of the field of computational linguistics". In: *Journal of the Association for Information Science and Technology* 67.3 (2016), pp. 683–706.

[57] Alec Radford et al. "Improving language understanding by generative pre-training". In: (2018).

[58] Colin Raffel et al. "Exploring the limits of transfer learning with a unified text-to-text transformer." In: *J. Mach. Learn. Res.* 21.140 (2020), pp. 1–67.

[59] Marc'Aurelio Ranzato et al. "Sequence level training with recurrent neural networks". In: *arXiv preprint arXiv:1511.06732* (2015).

[60] Luz Rello et al. "Frequent words improve readability and short words improve understandability for people with dyslexia". In: *IFIP Conference on Human-Computer Interaction*. Springer. 2013, pp. 203–219.

[61] Steven J Rennie et al. "Self-Critical Sequence Training for Image Captioning". In: *CVPR*. 2017.

[62] RJ Senter and Edgar A Smith. *Automated readability index*. Tech. rep. Cincinnati Univ OH, 1967.

[63] Matthew Shardlow. "A survey of automated text simplification". In: *International Journal of Advanced Computer Science and Applications* 4.1 (2014), pp. 58–70.

[64] Advaith Siddharthan. "A survey of research on text simplification". In: *ITL-International Journal of Applied Linguistics* 165.2 (2014), pp. 259–298.

[65] Irena Spasic, Goran Nenadic, et al. "Clinical text data in machine learning: systematic review". In: *JMIR medical informatics* 8.3 (2020), e17984.

[66] Neha Srikanth and Junyi Jessy Li. "Elaborative Simplification: Content Addition and Explanation Generation in Text Simplification". In: *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*. 2021, pp. 5123–5137.

[67] Sanja Štajner. "Automatic Text Simplification for Social Good: Progress and Challenges". In: *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021* (2021), pp. 2637–2652.

[68] Renliang Sun, Hanqi Jin, and Xiaojun Wan. "Document-Level Text Simplification: Dataset, Criteria and Baseline". In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Online and Punta Cana, Dominican Republic: Association for Computational Linguistics, Nov. 2021, pp. 7997–8013. DOI: 10.18653/v1/2021.emnlp-main.630. URL: https://aclanthology.org/2021.emnlp-main.630.

[69] Sai Surya et al. "Unsupervised Neural Text Simplification". In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. 2019, pp. 2058–2068.

[70] Suha S Al-Thanyyan and Aqil M Azmi. "Automated text simplification: a survey". In: *ACM Computing Surveys (CSUR)* 54.2 (2021), pp. 1–36.

[71] S Rebecca Thomas and Sven Anderson. "WordNet-based lexical simplification of a document." In: *KONVENS*. 2012, pp. 80–88.

[72] Adel I Tweissi. "The effects of the amount and type of simplification on foreign language reading comprehension". In: (1998).

[73] Ashish Vaswani et al. "Attention is all you need". In: *Advances in neural information processing systems* 30 (2017).

[74] Guillaume Wenzek et al. "CCNet: Extracting High Quality Monolingual Datasets from Web Crawl Data". In: *Proceedings of the 12th Language Resources and Evaluation Conference*. 2020, pp. 4003–4012.

[75] Ronald J Williams. "Simple statistical gradient-following algorithms for connectionist reinforcement learning". In: *Machine learning* 8.3 (1992), pp. 229–256.

[76] Sander Wubben, Antal Van Den Bosch, and Emiel Krahmer. "Sentence simplification by monolingual machine translation". In: *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 2012, pp. 1015–1024.

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

[78] Wei Xu et al. "Optimizing Statistical Machine Translation for Text Simplification". In: *Transactions of the Association for Computational Linguistics* 4 (2016), pp. 401–415.

[79] Yu Yan et al. "FastSeq: Make Sequence Generation Faster". In: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th Inter-*

national Joint Conference on Natural Language Processing: System Demonstrations. 2021, pp. 218–226.

[80] Weizhe Yuan, Graham Neubig, and Pengfei Liu. "Bartscore: Evaluating generated text as text generation". In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 27263–27277.

[81] Torsten Zesch, Christof Müller, and Iryna Gurevych. "Extracting Lexical Semantic Knowledge from Wikipedia and Wiktionary". In: *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*. Marrakech, Morocco: European Language Resources Association (ELRA), May 2008. URL: `http://www.lrec-conf.org/proceedings/lrec2008/pdf/420_paper.pdf`.

[82] Jingqing Zhang et al. "PEGASUS: Pre-training with Extracted Gap-sentences for Abstractive Summarization". In: *International Conference on Machine Learning*. PMLR. 2020, pp. 11328–11339.

[83] Xingxing Zhang and Mirella Lapata. "Sentence Simplification with Deep Reinforcement Learning". In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. 2017, pp. 584–594.

[84] Sanqiang Zhao et al. "Integrating Transformer and Paraphrase Rules for Sentence Simplification". In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. 2018, pp. 3164–3173.

[85] Yanbin Zhao et al. "Semi-supervised text simplification with back-translation and asymmetric denoising autoencoders". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 34. 05. 2020, pp. 9668–9675.

[86] Zhemin Zhu, Delphine Bernhard, and Iryna Gurevych. "A monolingual tree-based translation model for sentence simplification". In: *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*. 2010, pp. 1353–1361.

Appendix A

Training Procedures and Decoding Methods

A.1 BART-UL

Devaraj et al. [18] proposed BART-UL, a model which uses the BART model as the backbone for paragraph-level medical text simplification. To ensure that the model does not generate technical words, Devaraj et al. [18] adapted the concept of unlikelihood training to penalize the model whenever it generated technical words. The proposed unlikelihood training objective (i.e., unlikelihood loss (UL)) [80] is calculated as follows and is shown in Equation A.1 [18]

$$UL = -\sum_{t=1}^{y}\sum_{j=1}^{S} \mathbb{1}_{s_j,t} w_j log(1 - p_\theta(s_j|y_{<t}, x)) \tag{A.1}$$

where S is a set of candidate tokens, x is the complex medical paragraph as the input, $y_{<t}$ is the prefix of simple medical paragraph y,and $p_\theta(s_j|y_t, x)$ is the probability assigned to the token s_j in the distribution output by BART with model parameters at time t [18]. The set

of candidate tokens (S) is calculated by collecting tokens with negative weights using a bag of word logistic regression which is trained to classify whether the given paragraph is simple or complex. The unlikelihood loss mentioned in equation A.1 is only applied for a given token s_j with learned logistic regression weight w_j if the output probability distribution of the BART model for the token indicates that it should be in the generated output. The final loss function for training BART-UL is the weighted sum of unlikelihood loss and the standard maximum likelihood loss. The proposed training method by Devaraj et al. [18] helped them perform paragraph-level simplification on medical text data.

A.2 MUSS: Multilingual Unsupervised Sentence Simplification by Mining Paraphrases

Martin et al. [41] recently introduced MUSS, a BART [14] based language model which achieved state-of-the-art results on TS benchmarks. MUSS was trained on a data set created by mining paraphrases from the CCNET corpus. This data set is a sentence-level data set i.e., data set contains complex sentences and corresponding simple sentences. During the training time, MUSS uses control tokens that tell the model about important properties of the target sentence. The control tokens used by MUSS are character length ratio (NumChar), replace-only Levenshtein similarity (LevSim), Word frequency ratio (WordFreq), Dependency Tree Depth Ratio (DepTreeDepth). These control tokens were first proposed by Martin et al. [40] and their importance is given below:

- Character Length Ratio: This control token measures compression and content deletion between source and target sentence. [40]
- Levenshtein Similarity: Levenshtein similarity measures the amount of modifications

done on source sentences via deletion, addition or replacement. In case MUSS they have only considered replacement as paraphrases often do not involve heavy deletion or addition operations. [40]

- Word Frequency Ratio: Word Frequency are shown to be good indicators of word complexity and hence word frequency ratio between source and target sentences serves as a proxy to measure lexical similarity. [40]

- Dependency Tree Depth Ratio (DepTreeDepth): Dependency Tree Depth Ratio is measured as maximum depth of dependency tree of the source sentence divided by that of the target sentence. DepTreeDepth serves as a proxy to syntactic simplicity. [40]

These four tokens are prepended to every source sentence while training the MUSS model. Overall, MUSS is trained on a data set consisting of one million paraphrases. In our experiments we did not further fine-tune the MUSS model as the control tokens designed are for sentence level simplification tasks whereas the data set proposed by Devaraj et al. [18] is a paragraph level text simplification data set and designing the oracle tokens for a paragraph level task is out of the scope for the current paper. Although MUSS is trained on sentence level data, it still serves as a strong baseline because of the huge corpus data that was used to train the model. Due to this reason, we have included the MUSS model as a baseline.

A.3 Keep it Simple: Unsupervised Simplification of Multi-Paragraph Text

Laban et al. [34] proposed Keep it Simple (KIS) an unsupervised reinforcement learning-based approach to simplify paragraph-level text. They propose a variant of SCST [65] called K-SCST in which instead of proposing one candidate simplification, the model proposes multiple candidate simplifications, computes the reward for candidates and encourages simplification which outperforms the mean reward. More formally in K-SCST, k sampled sentences are generated and rewards are computed for each candidate $R^{S_1}, \cdots, R^{S_k}$ and the baseline is approximated as the mean of these rewards (R^S). The loss function is defined as follows and shown in Equation A.2 [7]

$$L = \sum_{j=1}^{k} (R^S - R^{S_j}) \sum_{i=0}^{N} logp(w_i^{S_j} | w_i^{S_j} \cdots w_{i-1}^{S_j}, P) \tag{A.2}$$

Where P is the input sentence and L is the resulting loss function and k is the number of sampled sentences. The rewards used by KIS are Salience Rewards, Lexical Simplicity Rewards, Syntactic Simplicity Rewards, and Language model-based Fluency Reward. All the rewards are unsupervised i.e., they do not require any reference sentence and only require the source and generated sentences. They also have introduced guardrails to maintain length and accuracy of generated text. A GPT2 model is trained using KIS procedure on an unreleased data set of 7 million news articles. For our experiments, we fine-tuned the GPT2 model using KIS procedure on the data set proposed by Devaraj et al. [18]. Since KIS is an unsupervised TS approach, it requires a lot of data to reach an optimal score. Unfortunately, the data set released by Devaraj et al. [18] has only 3568 training instances and hence is not enough to ensure that model can be trained properly. We believe that having a large corpus of

paragraph level medical text can help in stabilizing the training procedure. For more details about the rewards, data sets used in KIS procedure readers are suggested to refer to [34].

A.4 Decoding Strategies

Greedy Decoding

The transformer based language models which are used in sequence to sequence tasks usually follow an encoder-decoder structure. The encoder side takes an input sequence and outputs a continuous sequence of representations). The decoder takes these continuous representations as input and outputs a generated sequence . At each generation step the model is autoregressive, i.e., it consumes previously generated tokens and outputs the probability scores to select next tokens. In greedy decoding, the token with maximal probability is always selected. In general, greedy decoding step at time t is denoted as follows

$$\hat{w_t} = \underset{w_t}{\operatorname{argmax}}(p(w_t|z_t)) \tag{A.3}$$

Where denotes the next generated token, denotes previous generated token and denotes representation of input tokens obtained from encoder [79].

Appendix B

Hyperparameters and Evaluation Metrics

B.1 TESLEA: Hyper-Parameter Settings

The data-split used is the same as proposed by Devaraj et al. [18] with 3568 reviews in the training set, 411 in the validation set, and 480 in the test set. The pretrained BART model is initialized from a checkpoint trained on the XSum data set [42]. The model parameters are updated using AdamW [82], with a learning rate of 2e-5 for both initial fine-tuning and RL training. The model was first fine-tuned for 10 epochs and then trained for 30 epochs using RL training. From experiments performed on the validation set, we found that the optimal value for the scaling factor is 0.95. We equally weighted all the rewards as they encapsulate all the properties required for a good simplification. All experiments were performed on a single NVIDIA A-100 GPU with a memory size of 40GB. We mainly experimented with 3 variants of the BART-model, namely BART-base, BART-large and BART-large-Xsum. The variants differed in the model size, pre-training data and Batch Size every other parameter

was kept constant to stabilize the training regime. Table B.1 describes the variants of BART model along with model size, pre-training data, batch size, Time to train (reported in days), Inference speed per sample on Test Data set (reported in seconds) and the FKGL score obtained by each model. One can observe from Table B.1 that the BART-large-xsum variant performs the best on FKGL score.

Model	**Model Size**	**Pre-Training Data**	**Batch Size**	**Time to Train**	**Inference Speed**	**FGKL Score**
Bart-Base	139 M	English Wikipedia + Book Corpus	2	4 days	1.3s	13.23
Bart-Large	406 M	English Wikipedia + Book Corpus	1	7 days	1.75s	13.48
Bart-Large-Xsum	406 M	English Wikipedia + Book Corpus + Xsum	1	7 days	1.75s	**11.84**

Table B.1: Information about BART-variants and parameters. Time to train is measured in days and Inference speed is measured in seconds.

B.2 Automatic Evaluation Metrics

BARTScore

Yuan, Neubig, and Liu [80] framed the problem of evaluating generated text as a text generation problem. BARTScore helps to assess the quality of the generated text. They evaluate generated text via the probability of it being generated from other text (ie, source texts or reference texts) or vice versa. The BART [35] model is used to estimate the probabilities required to calculate the given scores. Given one text y and another text x, BARTScore is

calculated using weighted probability and is calculated by Equation B.1 below [80]:

$$\text{BARTScore} = \sum_{i=1}^{m} w_t log p(y_t|y_{<t}, x, \theta) \tag{B.1}$$

where m is the length of y; denotes preceding tokens before the position t; are model parameters and are the weights associated with different tokens, however Yuan, Neubig, and Liu [80] weigh each token equally. The criteria proposed by Yuan, Neubig, and Liu [80] to measure the quality of the generated text is given below :

- Informativeness (INFO): Does the generated text capture the important ideas of the source text [80].
- Fluency (FLU): Does the generated text has no formatting problems or grammatical errors that increase the difficulty to read the text. [80].
- Coherence (COH): Whether the generated text relates from sentence to sentence in a logically consistent order to present information about a topic [80].
- Factuality (FAC): Whether the generated text contains only statements supported by the source text (i.e., no new information is being introduced) [80].
- Adequacy (ADE): Whether the generated text conveys the same meaning as the source text, and none of the important information is missing or added or misreported [80].

Yuan, Neubig, and Liu [80] also introduced four different settings for the evaluation of the criteria mentioned above. For a given source text (s), generated text (h), and reference text (r), the settings are defined as follows:

- Faithfulness Score ($s \rightarrow h$): This score measures how likely it is that generated text can be obtained given the source text. Faithfulness score can be used to measure factors of coherence, fluency, factuality, and relevance.

- Precision ($r \rightarrow h$): This score measures how likely generated text can be obtained from reference texts.
- Recall ($h \rightarrow r$): This score measures how likely generated reference texts can be obtained from the generated text.
- F-score ($r \rightarrow$): F-score is the average of precision and recall scores and can be used to measure adequacy and informativeness.

B.3 Abbreviations

Description	**Abbreviation**
Text Simplification	TS
Reinforcement Learning	RL
Self Critical Sequence Training	SCST
Flesch-Kincaid Grade Level	FKGL
Recall-Oriented Understudy for Gisting Evaluation	ROUGE
Fluency	FLU
Coherence	COH
Factuality	FAC
Informativeness	INFO
Adequacy	ADE
Average Likert Scores	ALS
Google Scholar	GS
Natural Language Generation	NLG
Natural Language Processing	NLP

Table B.2: List of Abbreviations

B.4 Code

All the code is open sourced. The code for TESLEA can be found here and code for Biblometric analysis can be found here.