NapSS: Paragraph-level Medical Text Simplification via Narrative Prompting and Sentence-matching Summarization

Junru Lu[1], Jiazheng Li[2], Byron C. Wallace[3], Yulan He[1,2,4] and Gabriele Pergola[1]

[1]Department of Computer Science, University of Warwick, UK
[2]Department of Informatics, King's College London, UK
[3]Northeastern University, USA   [4]The Alan Turing Institute, UK
{junru.lu, gabriele.pergola}@warwick.ac.uk
b.wallace@northeastern.edu, {jiazheng.li, yulan.he}@kcl.ac.uk

Abstract

Accessing medical literature is difficult for laypeople as the content is written for specialists and contains medical jargon. Automated text simplification methods offer a potential means to address this issue. In this work, we propose a *summarize-then-simplify* two-stage strategy, which we call NapSS, identifying the relevant content to simplify while ensuring that the original narrative flow is preserved. In this approach, we first generate reference summaries via sentence matching between the original and the simplified abstracts. These summaries are then used to train an extractive summarizer, learning the most relevant content to be simplified. Then, to ensure the narrative consistency of the simplified text, we synthesize auxiliary *narrative prompts* combining key phrases derived from the syntactical analyses of the original text. Our model achieves results significantly better than the seq2seq baseline on an English medical corpus, yielding 3%~4% absolute improvements in terms of lexical similarity, and providing a further 1.1% improvement of SARI score when combined with the baseline. We also highlight shortcomings of existing evaluation methods, and introduce new metrics that take into account both lexical and high-level semantic similarity. A human evaluation conducted on a random sample of the test set further establishes the effectiveness of the proposed approach. Codes and models are released here: https://github.com/LuJunru/NapSS.

1 Introduction

The medical literature is vast, and continues to expand quickly. Most patients (laypeople), however, are unable to access this information because it is written for specialists and so dense and laden with jargon. As the recent 'infodemic' has shown, access to reliable and comprehensible information about citizens' health is a fundamental need: for example, a European Health Literacy Survey (HLS-EU) reports that "at least 1 in 10 (12%) respondents

Figure 1: A typical sample of Medical Text Simplification task. The abstract and plain-language summary are split into sentences for easy inspection. Key phrases in each sentence, and marks of chosen sentences in reference summary are in bold.

show insufficient health literacy and almost 1 in 2 (47%) has insufficient or problematic health literacy" (Sørensen et al., 2015). Automated text simplification methods offer a potential means to address this issue, and make evidence available to a wide audience as it is published. However, performing paragraph-level simplification of medical texts is a challenging NLP task.

Online medical libraries such as Cochrane library,[1] provide synopses of the medical literature across diverse topics, and manually-written plain language summaries. We are interested in developing accurate automated medical text simplification systems upon those libraries to help timely popularization of medical information to lay audience.

---

[1]https://www.cochranelibrary.com/

We show a typical example of a technical abstract and associated simplified summary from a recently introduced paragraph-level medical simplification corpus (Devaraj et al., 2021a) in Figure 1. The sample consists of a technical abstracts (ABS) written for experts, and an manually authored Plain-Language Summaries (PLS) of the same publication collected from the Cochrane website. The dataset only provide raw abstract-PLS pairs. For easy inspection, we further add sentence splitting and highlight key phrases.

As this example illustrates, a text simplification system needs to first have an overview of the key details reported in the abstract (e.g., that the review synthesizes '*two trials*') and must also infer that there '*were no significant differences*' when '*rapid negative pressure application*' was applied to all participants, and thus that the '*rapid method should be recommended*'. This entails an overall understanding of the key concepts to simplify, while preserving a consistent narrative flow. Built upon this general framing, the system should identify that the most representing sentences in the abstract are sentences 1, 6, 5 and 7. The key challenges here for a model include: (i) identifying the most important content to simplify within the synopsis; (ii) preserving the original narrative flow from a linguistic and medical point of view; (iii) synthesising the findings in a simple and consistent language.

To address these challenges, we propose a *summarize-then-simplify* two-stage framework NapSS—**Na**rrative **P**rompting and **S**entence-matching **S**ummarization—for paragraph-level medical text simplification. The *narrative prompt* is designed to promote the factual and logical consistency between abstracts (ABSs) and PLSs, while the *simplification-oriented summarizer* identifies and preserves the relevant content to convey and simplify.

In the first stage, we construct intermediate summaries via sentence matching between the abstract and the PLS sentences based on their Jaccard Distance. This preliminary set of summaries is used to fine-tune a simplification-oriented summarizer which at inference time identifies and extracts the most relevant content to be simplified from the technical abstracts. This extractive summarizer is simplification-aware in that the reference summary is built with PLS ground truth.

In the second stage of simplification, the intermediate summary is concatenated to a narrative prompt generated by synthesising the main concepts, entities, or events mentioned in text resulting from the syntactic analysis of the PLSs. The prepared input is passed to a seq2seq model (e.g., BART (Lewis et al., 2020)) to produce a plain-language output.

Our contributions can be summarized as follows:

- We introduce NapSS, a two-stage *summarize-then-simplify* approach for paragraph-level medical text simplification, leveraging extractive summarization and narrative prompting.

- We design a *simplification-aware summarizer* and a narrative prompt mechanism. The former is based on a Pre-trained Language Model (PLM) fine-tuned for extractive summarization on an intermediate set of summaries built via sentence matching between the technical and simplified text. The latter synthesises key concepts from the medical text by syntactic dependency parsing analyses, promoting the overall consistency with the narrative flow.

- We conduct a thorough experimental assessment on the Cochrane dataset for paragraph-level medical simplification, evaluating the different features of the generated text (i.e., simplicity and semantic consistency) using several automatic metrics, and the model generalization on sentence-level simplification. Additionally, to mitigate the limitations of the automatic metrics, we designed and conducted a human evaluation assessment, involving "layperson" readers and medical specialists. The results demonstrated the state-of-the-art performance on quality and consistency of the simplified text.

2 Related work

We review three lines of work relevant to this effort: text simplification, extractive summarization, and prompting.

2.1 Text Simplification

Work on text simplification has mainly focused on sentence-level simplification, using the Wikipedia-Simple Wikipedia aligned corpus (Zhu et al., 2010; Woodsend and Lapata, 2011) and the Newsela simplification corpus (Xu et al., 2015). There has been less work on document-level simplification, perhaps owing to a lack of resources (Sun et al., 2021; Alva-Manchego et al., 2019).

The medical domain stands to benefit considerably from automated simplification: The medical literature is vast and technical, and there is a need to make this accessible to non-specialists (Kickbusch et al., 2013). Some research uses those medical documents and deploys various simplification methods based on lexical and syntactic simplification (Damay et al., 2006; Kandula et al., 2010; Llanos et al., 2016). The recent release of the Cochrane dataset provided a new parallel corpus of technical and lay overview of published medical evidence (Devaraj et al., 2021a).

2.2 Extractive Summarization

Extractive summarization aims to select the most important words, sentences, or phrases from input texts and combine them into a summary. Many approaches have been proposed: ranking and selecting sentences based on their graph overlap (Mihalcea and Tarau, 2004), deriving the relevance of the sentences within the text using WordNet (Pal and Saha, 2014), extracting information by named entity recognition (Maddela et al., 2022), and using continuous vector representations to perform semantic matching and sentence selection (Liu and Lapata, 2019; Narayan et al., 2018b; Gui et al., 2019; Lu et al., 2020; Pergola et al., 2021a).

There are some works that focus on extractive summarization of biomedical texts (Mishra et al., 2014; Sun et al., 2022). These have either aimed to provide a summary via graph-based methods or via sequence extraction to present key information in structured (tabular) form (Gulden et al., 2019; Aramaki et al., 2009). In this work we follow a standard *sentence matching* extractive summarization method (Goldstein et al., 1999; Zhong et al., 2020) and fine-tune a pre-trained language model to perform sentence classification. We use extractive summaries as an intermediate step.

2.3 Prompting

Recent work has shown that language models can be *prompted* to perform tasks without supervision (i.e., "zero-shot") (Radford et al., 2018; Brown et al., 2020). Prompts have been shown to work across a wide range of NLP tasks, e.g., sentiment classification, "reading comprehension", and "commonsense reasoning" (Seoh et al., 2021; Petroni et al., 2019; Pergola et al., 2021b; Jiang et al., 2019; Lu et al., 2022; Zhu et al., 2022; Wei et al., 2022). Recent work has shown that prompt-based methods can be used even with smaller language models

(Schick and Schütze, 2020; Gao et al., 2020). In this work we focus on a novel use of prompts: Assisting generation of simplified text.

3 Methods

We first define the Paragraph-level Text Simplification task, introducing the relevant notations, and then present the NapSS model.

3.1 Task Formulation

In many cases, text simplification can be viewed as a generative task with additional constraints regarding the simplicity of the generated text. Analogously to text summarization, paragraph-level text simplification can be formulated as follows: for a given *complex* paragraph with M sentences, $\mathbf{x} = \{\{x_1^1, x_2^1, \cdots, x_{N_{x1}}^1\} \cdots \{x_1^M, x_2^M, \cdots, x_{N_{xM}}^M\}\}$, the aim is to generate a *plain-language summary* (PLS) $\hat{\mathbf{y}} = \{\hat{y}_1, \hat{y}_2, \cdots, \hat{y}_{N_s}\}$, summarizing and simplifying the original paragraph, with N_{x^m} denoting the length of the m-th sentence $\mathbf{x^m}$.

3.2 NapSS

We now describe NapSS, a text simplification approach based on a *summarize-then-simplify* two-stage pipeline with the aims of (i) identifying the relevant content to simplify while (ii) ensuring that the original narrative flow is preserved. First, we generate a preliminary summary by using a *simplification-oriented BERT summarizer*, an extractive model fine-tuned beforehand to identify the most relevant content to attend and simplify (§3.2.1). These preliminary summaries are then combined with a *narrative prompt*, a synthetic set of key phrases describing the main concepts, entities, or events discussed in the original text and derived from its syntactic analysis (§3.2.2). The overall working flow of our proposed NapSS model is illustrated in Figure 2. We next provide the details of each of these modules.

3.2.1 Sentence-matching Summarization

The idea behind the summarization stage is to identify the most important content within a given technical abstract (with respect to target simplifications). We automatically construct an intermediate "reference" summary dataset using the simplification training set with which to fit a simplification-oriented summarizer. Specifically, we train the latter as a binary sentence classifier, which provides a simple extractive summarization approach.
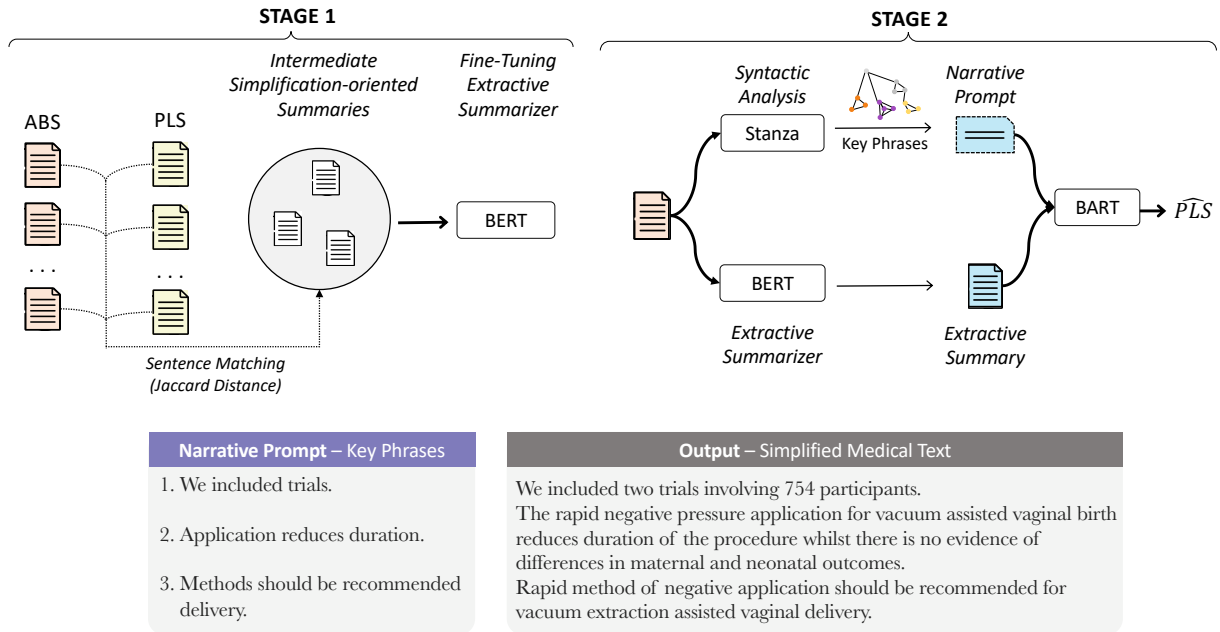
STAGE 1

Intermediate Simplification-oriented Summaries

Fine-Tuning Extractive Summarizer

ABS   PLS

. . .   . . .

Sentence Matching
(Jaccard Distance)

BERT

STAGE 2

Syntactic Analysis

Narrative Prompt

Stanza

Key Phrases

BART → $\hat{PLS}$

BERT

Extractive Summarizer

Extractive Summary

Narrative Prompt – Key Phrases	Output – Simplified Medical Text
1. We included trials.	We included two trials involving 754 participants.
2. Application reduces duration.	The rapid negative pressure application for vacuum assisted vaginal birth reduces duration of the procedure whilst there is no evidence of differences in maternal and neonatal outcomes.
3. Methods should be recommended delivery.	Rapid method of negative application should be recommended for vacuum extraction assisted vaginal delivery.

Figure 2: Overview of the two-stage pipeline in the NapSS model. In the first stage, we perform sentence "labelling" using Jaccard Distances (Jaccard, 1912) over abstract (ABS) sentences in reference to PLS sentences, generating a set of intermediate summaries. A binary BERT-based (Devlin et al., 2018) classifier is fine-tuned over these summaries and used, at test time, to generate an extractive summary $\mathbf{x}'$. During the second stage (right side), we perform syntactic dependency parsing over the PLS sentences to extract key phrases $\mathbf{k}$. These are concatenated to form a narrative prompt and combined with the extractive ABS summary to serve as input of the simplification module for the generation of plain-language outputs $\hat{\mathbf{PLS}}$. In the bottom part, we reported an example of narrative prompt and simplified text generated by NapSS on the ABS introduced in Figure 1.

Algorithm 1 details the process of building this pseudo reference summary dataset. The input to the algorithm are the sets of sentences from the technical abstract (ABS) and the corresponding simplified text (PLS). For each PLS sentence, we calculate the Jaccard Distance to every ABS sentence, and select the one with the lowest score. The set of selected ABS sentences constitute an intermediate extractive summary of the technical abstract. The complexity of Algorithm 1 is $O(N_x \cdot N_y \cdot D)$, where D denotes the size of entire corpus.

Based on the intermediate summary dataset, we fine-tune a BERT model to perform binary classification over sentences. At inference time, the resultant trained *simplification-oriented summarizer* is used to select sentences from the technical abstract which will be simplified. These are concatenated and then passed to a BART model (Lewis et al., 2020) along with the narrative prompt.

As an example, the bottom left of Figure 2 shows 3 PLS sentences guiding the automatic labelling (0/1) of 7 ABS sentences. The intermediate extracted summary $\mathbf{x}'$ derived via Jaccard matching is used at training time, while at inference time we extract this using the trained model.

3.2.2 Narrative Prompting

Intuitively, the simplification-oriented summarizer should identify the most important content in ABS which should be simplified. However, the similarity matching with which we train the sentence classifier may be noisy and miss relevant information constituting the narrative flow, resulting in errors that lead to omissions in outputs. Therefore, in our NapSS model, we incorporate another simple mechanism, *narrative prompting*, to encourage factual consistency between the input and output.

Inspired by recent work on *chain-of-thought* "reasoning" (Wei et al., 2022), we assume a logical narrative chain can be explicitly constructed with key phrases extracted via syntactic dependency parsing, and then used as a prompt. Specifically, we use a light natural language processing tool Stanza[2] for dependency parsing on every abstract sentence to extract key phrases. Algorithm 2 details the algorithmic process of our narrative prompting. The algorithm takes abstract sentences as input, runs a dependency parse on each, collects the root token and its closest child tokens to form key phrases in

---

[2] https://stanfordnlp.github.io/stanza/

Algorithm 1 Build reference summary dataset
1: Input require: abstract sentence sets $\{\mathbf{x^m_{1\sim M}}\}$,
2: PLS sentence sets $\{\mathbf{y^q_{1\sim Q}}\}$
3: Initialization: Empty positive sentence set x_{pos}
4: **for** PLS sentence $\mathbf{y^q} \in \{\mathbf{y^q_{1\sim Q}}\}$ **do**
5:    Initilization: Minimum Jaccard Distance $\text{Dist}_q \leftarrow 10.0$,
6:    corresponding sentence index $\text{Ind}_q \leftarrow 0$
7:    **for** abstract sentence $\mathbf{x^m} \in \{\mathbf{x^m_{1\sim M}}\}$ **do**
8:       $\text{Dist}_{qm} = \texttt{JaccardDistance}(\mathbf{y^q}, \mathbf{x^m})$
9:       **if** $\text{Dist}_{qm} < \text{Dist}_q$ **then**
10:          $\text{Dist}_q \leftarrow \text{Dist}_{qm}$
11:          $\text{Ind}_q \leftarrow m$
12:       **end if**
13:    **end for**
14:    **if** $\mathbf{x}^{\text{Ind}_q} \notin x_{pos}$ **then**
15:       add $\mathbf{x}^{\text{Ind}_q}$ in x_{pos}
16:    **end if**
17: **end for**
18: Negative sentence set $x_{\text{neg}} = \{\mathbf{x^m_{1\sim M}}\} - x_{\text{pos}}$

Algorithm 2 Build narrative prompt
1: Input require: abstract sentence sets $\{\mathbf{x^m_{1\sim M}}\}$
2: Initilization: Empty key phrases queue x_{que}
3: **for** abstract sentence $\mathbf{x^m} \in \{\mathbf{x^m_{1\sim M}}\}$ **do**
4:    DTree = DependencyParsing($\mathbf{x^m}$)
5:    $\mathbf{x^m_{root}}$ = DTree.Root()
6:    $\mathbf{x^m_{root_l}}, \mathbf{x^m_{root_r}}$ = DTree.Children($\mathbf{x^m_{root}}$)
7:    $k^m = \mathbf{x^m_{root_l}} \, \mathbf{x^m_{root}} \, \mathbf{x^m_{root_r}}$
8:    add k^m in x_{que}
9: **end for**
10: Prompt $k^M = k^1$</s>k^2</s>$\cdots$</s>k^m

---

natural linguistic orders, and assembles these as the narrative prompt. Let k^m denotes the key phrase of sentence $\mathbf{x^m}$, the narrative prompt k^M equals to $[k^1$</s>k^2</s>$\cdots$</s>$k^m]$, in which "</s>" is a special separation token. The complexity of this building algorithm is $O(N_x \cdot D)$. As shown in Figure 2, key tokens are shown with bold fonts in every abstract sentences.

3.2.3 Text Simplification

The resulting input of the second text simplification stage is composed by $[k^M$</s>$\mathbf{x'}]$, as depicted in the bottom right part of Figure 2. NapSS adopts encoder-decoder PLM models as the backbone for generative text simplification. Let $L_{gen_{TS}}$ be the

loss of the generative text simplification task:

$$L_{gen_{TS}} = -\frac{1}{N_k + N_{x'}} \sum_{t=1}^{N_k+N_{x'}} y_t \log \hat{y}_t \quad (1)$$

where N_k, $N_{x'}$ are the lengths of the narrative prompt k^M and of the extractive summary $\mathbf{x'}$, respectively.

4 Experimental Assessment

4.1 Experimental Setup

Dataset We build and evaluate NapSS on the first published paragraph-level medical text simplification dataset (Devaraj et al., 2021a). The dataset is derived from the Cochrane library of systematic reviews and contains 4,459 parallel pairs of technical (ABS) and simplified (PLS) medical abstracts curated by domain experts. The average length of abstract is around 300 to 700 tokens, while the average length of PLS is around 130 to 390 tokens (Devaraj et al., 2021a). All abstract and PLS text are preprocessed to have a total token length lower than 1,024, which is a typical input upper bound of large PLM models. The dataset was split into 3,568 training, 411 development and 480 testing instances. To our knowledge, this is the only accessible paragraph-level text simplification dataset.

For the summarization model, the derived summary dataset contains 51,635 training, 5,856 development, and 7,009 testing sentences (constructed from the respective dataset splits). This dataset contains around 53% positive sentences and 47% negative sentences, which is relatively balanced, and consistent with the proportion of average amount of PLS sentences and average amount of paired abstract sentences. We describe hyperparameter selection in the Appendix Section A.1.

Evaluation Metrics For evaluation we largely adopt the metrics used in prior work on this task and dataset (Devaraj et al., 2021a). These can be placed into three groups: readability metrics, lexical similarity metrics, and simplification metrics. The readability metrics include the Flesch–Kincaid grade level score (FK) (Kincaid et al., 1975) and the automated readability index (ARI) (Senter and Smith, 1967). Lexical similarity metrics are widely adopted to evaluate text generation, including ROUGE-1, ROUGE-2, ROUGE-L (Lin, 2004) and BLEU (Papineni et al., 2002). The simplification metrics include SARI (Xu et al., 2016), which is an editing-base metric especially designed for

Models	Readability		Lexical Similarity				Simplification	Semantic Similarity	Comprehensive
	FK	ARI	Rouge-1	Rouge-2	Rouge-L	BLEU	SARI	BertScore	BLEURT
Vanilla BART	10.89	14.32	46.79	19.23	43.55	11.5	38.72	23.94	-0.194
UL-BART (Devaraj et al., 2021a)	11.97	13.73	38.00	14.00	36.00	39.0	40.00	/	/
UL-BART (by us)	9.30	12.40	43.25	16.36	40.22	7.9	40.08	24.64	-0.309
NapSS (our)	10.97	14.27	**48.05**	19.94	44.76	**12.3**	40.37	**25.73**	**-0.155**
NapSS BioBART	10.98	14.24	47.66	19.77	44.39	11.9	40.21	25.61	-0.166
NapSS (+UL)	**8.67**	**11.80**	45.39	16.77	42.53	9.1	**41.12**	23.13	-0.219
NapSS (-Prompt)	9.86	13.06	45.62	**20.01**	**44.83**	12.1	39.68	25.57	-0.158
NapSS (-Summary)	10.62	13.99	46.91	19.51	44.18	11.8	39.62	25.29	-0.167

Table 1: Overall results on the testing set. UL BART is the previous SOTA, and we report results from our re-implementation of this. The inconsistency between (Devaraj et al., 2021a) and our re-implementation is due to the inavailability of evaluation code. For NapSS, we provide 2 groups of results by changing backbone model of text simplification module. The robustness verification of proposed NapSS is provided in appendix B. We further provide fusion and ablation results based on BART version of NapSS. NapSS (-Prompt) refers to remove the narrative prompt, while NapSS (-Summary) is to replace the abstract summary with full abstract.

text simplification task. In our setting, SARI would reward the *generation* of words occurring only in the paired PLSs, and avoidance of ABS words not occurring in the corresponding PLS.

Simple automated metrics fail to capture semantic agreement between outputs and references. We therefore consider two additional metrics: BertScore (Zhang et al., 2019) and BLEURT (Sellam et al., 2020). BertScore was originally designed to evaluate semantic similarity via BERT (Devlin et al., 2018) embeddings. Alva-Manchego et al. (2021) and Devaraj et al. (2022) recently assessed and verified its effectiveness on the text simplification task. BLEURT is a metric finetuned on both lexical BLEU metric and semantic BertScore metric. Along with the automatic assessment, we also conduct a manual (human) evaluation of the simplicity, fluency and factuality whose evaluation criteria are detailed Section §4.2.3.

Prior work did not publicly provide code to perform evaluations beyond computing ROUGE.[3] Therefore, we mainly compare results according to our re-implementation of evaluation metrics.

Baseline "Vanilla" BART is a pretrained encoder-decoder architecture, based on transformers, whose auto-regressive decoder made it a suitable a strong baseline for text generation. In ours setting, we adopted a a specific checkpoint version[4] additionally fine-tuned on the XSUM dataset (Narayan et al., 2018a; Devaraj et al., 2021b), providing higher performance on text summarization. The only other model developed for paragraph-level

medical text simplification is *UL-BART* (Devaraj et al., 2022), is also based on BART but integrates an auxiliary "unlikelihood" (UL) penalty to demote generation of technical jargon, which improved the readability and simplicity of outputs compared to the base BART model.

4.2 Results

4.2.1 Automatic Metrics

We report quantitative results in Table 1 comparing the main models and the ablation studies. We notice that UL-BART can generate text which is more readable (lower FK and ARI) and simpler (higher SARI) than the "Vanilla" BART. However, the model struggles to maintain lexical and semantic similarity (lower ROUGE, BLUE, and higher BLEURT) to the human references, perhaps because omitting jargon terms as the modified objective degrades coherence.

By contrast, NapSS improves lexical similarity by 3% to 4% in terms of ROUGE and BLEU scores while maintaining a comparable SARI score. NapSS additionally improves the semantic similarity between the model outputs and the human references at the cost of a slightly higher FK and ARI scores, demonstrating an higher semantic consistency while simplifying the medical text. For the sake of completeness, we also tested whether replacing the "Vailla" BART backbone with a specialised medical PLM, such as BioBART (Yuan et al., 2022), would lead to better performance. Surprisingly, the replacement did not lead to any significant change in any of the adopted metrics.

We further explored the integration of the auxiliary "unlikelihood" (UL) loss in NapSS (+UL), aiming at increasing the degree of simplification

---

[3] https://github.com/AshOlogn/Paragraph-level-Simplification-of-Medical-Texts
[4] https://huggingface.co/facebook/bart-large-xsum

Models	FK	ARI	BLEU	SARI	BLEURT
Vanilla BART	4.91	**6.83**	9.71	43.47	-0.663
UL-BART (by us)	**4.76**	7.61	8.75	40.83	-0.654
NapSS (our)	6.32	7.99	10.1	**45.78**	-0.648
NapSS (+UL)	5.49	8.25	**12.8**	44.46	**-0.553**

Table 2: Zero-shot inference results. All above models are only fine-tuned on the Cochrane dataset (2021a), then run zero-shot inference on the TICO-19 testing set.

Models	Simplicity	Fluency	Factuality	(Experts)	Overall
UL BART (by us)	**1.43**	1.53	1.17	0.99	4.13
NapSS	1.12	**1.54**	**1.66**	**1.28**	**4.32**

Table 3: Human evaluation result by each category.

while preserving semantic consistency. The resulting model yielded further state-of-the-art performance on the overall text simplicity with an increase of ~0.8% in readability and 1.1% in SARI score. NapSS (-Prompt) and (-Summary) refer to two ablation models. The first one removes the narrative prompt, leading to improved readability but decreased simplification (lower SARI). The second one show that the full abstract is necessary for improving the lexical similarity.

We report and discuss in Appendix C the binary classification performance of the extractive summarization module used in stage one.

4.2.2 Out-of-Domain Evaluation

To evaluate the generalization ability of NapSS, we evaluate the model on a *different* medical text simplification dataset: TICO-19 (Shardlow and Alva-Manchego, 2022). Unlike the Cochrane dataset, this is designed for sentence-level simplification and contains over 6k parallel technical and simplified sentences related to COVID-19.

Table 2 reports results. The "Vanilla" BART and UL-BART have the best performance on readability while NapSS yields over ~2% improvement in terms of simplicity. Integrating NapSS with the "unlikelihood" (UL) penalty (NapSS (+UL)) achieves around ~1-3% boost on lexical and semantic evaluation. The overall results highlight that our approach can preserve a high level of semantic consistency for simplification at the sentence level, yet with slightly reduced readability.

4.2.3 Human Evaluation

We designed and conducted a manual evaluation of the outputs generated by the simplification models to provide additional insights into *fluency* and *factuality*; the latter is especially difficult to assess with existing automatic metrics.

Evaluation Procedure We randomly sampled 100 unsimplified instances (ABSs) from the test set and paired each with simplified outputs gener-

ated by two models, one from UL-BART (Devaraj et al., 2021a) and one from the proposed NapSS. Each simplified text was assessed by three different annotators. We hired 6 annotators to participate in this evaluation, who are postdoctoral researchers and PhD students in computer science. Each was assigned 100 instances; this took nearly 8 hours to complete. Additionally, we hired two expert annotators who have professional background in the medical domain to obtain a reliable evaluation on the factual consistency between the complex and the simplified text. Annotators were paid $19 per hour. To ensure that annotators shared a common understanding of our evaluation criteria, we held a tutorial session with detailed instructions and provided 20 instances as a trial run. We then resolved any annotation inconsistencies afterwards.

Evaluation Criteria We followed a previous approach to ask annotators to give numerical scores for each instance (Alva-Manchego et al., 2021). Considering the requirement for the simplification tasks and text styles characterizing medical documents (Devaraj et al., 2022), we separated numerical scores into three aspects: simplicity, fluency and factuality. Annotators can select a numerical rating (from 0, 1, and 2) for each aspect. Appendix A.2 provides details for each category.

Results In Table 3, we present average annotator scores assigned to all aspects. Our model achieves higher overall and average scores on Fluency and Factuality, respectively. UL-BART model got higher score on Simplicity because this model sometimes generates too simple outputs. Simplicity from our evaluation schema only focuses on evaluating the length of the text and the vocabulary. It does not involve the evaluation of the content. Therefore, if the generated text only contains a conclusion from the paragraph, our evaluator would give a higher score on Simplicity. On the contrary, the fluency and factuality aspects focus on evaluation at the context and semantic level, where our model got a higher score in the assessment. As Factuality is an aspect that the evaluation is subject to evaluators' background knowledge, therefore we selected those instance been given three different

Complex Medical Abstract – ABS

S1. Five randomized studies involving 1382 patients were included in this review.
S2. All the included studies involved advanced (T3 or T4) prostate cancer, had relatively small populations, and were of short duration.
S3. Few events were reported and did not assess disease-specific survival or metastatic disease. Only one study (N = 77) evaluated biochemical outcomes.
S4. A subgroup analysis found no significant differences in biochemical progression (defined by the authors as PSA ≥ 10 ng/mL) between IAS and CAS for Gleason scores 4 - 6, 7, and 8 - 10.
S5. For patients with a Gleason score > 6, reduction in biochemical progression favoured the IAS group (RR 0.10, 95% CI 0.01 to 0.67, P = 0.02).
S6. Studies primarily reported on adverse events.
S7. One trial (N = 43) found no difference in adverse effects (gastrointestinal, gynecomastia and asthenia) between IAS (two events) and CAS (five events), with the exception of impotence, which was significantly lower in the IAS group (RR 0.72, 95% CI 0.56 to 0.92, P = 0.008).
S8. Data from RCTs comparing IAS to CAS are limited by small sample size and short duration.
S9. There are no data for the relative effectiveness of IAS versus CAS for overall survival, prostate cancer-specific survival, or disease progression.
S10. Limited information suggests IAS may have slightly reduced adverse events.
S11. Overall, IAS was also as effective as CAS for potency, but was superior during the interval of cycles (96%).

Plain-Language Summary – PLS – Reference

S1. Five studies involving 1382 patients were included in this review.
S2. All the included studies involved advanced (T3 or T4) prostate cancer. No study was of adequate size and duration.
S3. Few events were reported and they did not assess disease-specific survival or metastatic disease.
S4. Only one study evaluated biochemical outcomes. Studies primarily reported on adverse events.
S5. There are no data for the relative effectiveness of IAS versus CAS for overall survival, prostate cancer specific survival, disease progression, or quality of life.
S6. Limited information suggests IAS may have slightly reduced adverse events.
S7. **In Hering 2000, IAS (18/25 versus 18/18) appears to be slightly more favorable than CAS in controlling impotence.**
S8. Overall, IAS was also as effective as CAS for potency, but was superior during the interval of cycles (96%).
S9. More research is needed.

Plain-Language Summary – PLS – UL+BART

S1. This review included five randomized controlled trials involving 1382 patients.
S2. All the included studies involved advanced prostate cancer, had relatively small numbers of patients, and were of short duration.
S3. Few side effects were reported and did not assess disease-specific survival or metastatic disease.
S4. Only one of the studies evaluated biochemical progression, and found that IAS was as effective as CAS for potency, but was better during the interval of cycles.
S5. Limited information suggests IAS may have slightly reduced side effects, **with the exception of impotence, which was lower in the IAS group.**

Plain-Language Summary – PLS - NapSS

S1. Five randomised controlled trials involving 1382 patients were included in this review.
S2. All the included studies involved advanced (T3 or T4) prostate cancer, had relatively small populations, and were of short duration.
S3. There are no data for the relative effectiveness of IAS versus CAS for overall survival, prostate cancer-specific survival, or disease progression.
S4. Overall IAS was also as effective as CAS in reducing the risk of adverse events.

Figure 3: Case study and error analysis on a typical example from the testing set. Smeared sentences illustrate factual improvement by NapSS, while underlined parts reveal information omission of our model outputs.

scores from basic evaluators to create an experts set. We can see experts' evaluation also shows the same trend. We believe the narrative prompt benefits this improvement. Our model tends to produce a reasonable reduction in the context while keeping the majority of critical points. It is also useful for the model to calibrate grammar and plausibility with prompts. Combined with narrative prompt, NapSS generates simplification more consistent with the original text than the UL-BART. We can observe the better performance on human evaluation results also correlated with the improvement in semantic and comprehensive metrics, which proves the necessity of semantic level simplification evaluation.

4.2.4 Case Study and Error Analysis

We present a case study and error analysis based on the examples reported in Figure 3.[5] The Abstract (ABS) mentions the the analysis of 5 studies on the effects of the continuous (CAS) or intermittent (IAS) androgen suppression therapy on advanced prostate cancer. The UL-BART model generated a slightly longer simplified text than NapSS. Specifically, sentence 4 from the UL-BART output mixed and linked the biochemical progression assessment with the IAS and CAS side-effect for potency. In contrast, sentence 3 generated by NapSS is more

relevant to the findings of all studies considered.

On the other hand, the last sentence 5 from the UL output reported a meaningful finding in consistence with reference sentence 7 from the PLS and reference sentence 7 from the ABS. NapSS instead omitted this information, probably because the related PLS sentences were not considered sufficiently relevant by the model.

5 Conclusions

We proposed a *summarize-then-simplify* two-stage model—NapSS—for paragraph-level medical text simplification. The first component is a "simplification-oriented" summarizer, which we trained over a heuristically derived set of "psuedo" references derived via sentence matching. At inference time, the summarizer extracts the most relevant content to be simplified. This is combined with an additional "narrative prompt" intended to promote consistency, and then passed to an encoder-decoder model to produce the simplified text. Experiments on a paragraph-level medical text simplification showed that, under several automatic metrics and human evaluation (involving "laypeople" and medical specialists), this method realized significant improvements with respect to both simplification quality and consistency.

---

[5]Better with colors

Limitations

Our study is primarily based on the Cochrane paragraph-level medical text simplification dataset (Devaraj et al., 2021a). While this dataset provides richer and more elaborated text than previous sentence-level medical datasets, such as TICO-19 (Shardlow and Alva-Manchego, 2022), it is worth noting that experimental documents tend to share a common pattern whose structure consists of: (i) discussing statistics about the clinical trials considered, (ii) list the experimental assessments, (iii) summarize the conclusions of the related findings.

Despite the already significant difficulty of the task, a limited variety of documents would inevitably introduce linguistic bias, hindering the model generalization and our current ability to conduct thorough assessment of the methodologies.

Moreover, although we made effort to examine the factuality aspect with expert annotators, we acknowledge that factuality is a subjective aspect and existing methods may not be sufficient to verify.

Ethics Statement

This work is based on publicly available medical datasets (Devaraj et al., 2021a; Shardlow and Alva-Manchego, 2022). As stated by the authors of datasets, no personal identification information were released. Current language technologies generally—and automated simplification models such as the one proposed in this work—still introduce "hallucinations" and factual inaccuracies into outputs; at present we would therefore recommend against deploying fully automated generative models for medical texts.

Acknowledgment

References

Fernando Alva-Manchego, Carolina Scarton, and Lucia Specia. 2019. Cross-sentence transformations in text simplification. In *Proceedings of the 2019 Workshop on Widening NLP*, pages 181–184, Florence, Italy. Association for Computational Linguistics.

Fernando Alva-Manchego, Carolina Scarton, and Lucia Specia. 2021. The (Un)Suitability of Automatic Evaluation Metrics for Text Simplification. *Computational Linguistics*, 47(4):861–889.

Eiji Aramaki, Yasuhide Miura, Masatsugu Tonoike, Tomoko Ohkuma, Hiroshi Masuichi, and Kazuhiko Ohe. 2009. Text2table: Medical text summarization system based on named entity recognition and modality identification. In *Proceedings of the BioNLP 2009 Workshop*, pages 185–192.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Jerwin Jan S Damay, Gerard Jaime D Lojico, Kimberly Amanda L Lu, and Dex B Tarantan. 2006. Simtext: text simplication of medical literature.

A. Devaraj, I. Marshall, B. Wallace, and J. J. Li. 2021a. 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*.

Ashwin Devaraj, Iain Marshall, Byron Wallace, and Junyi Jessy Li. 2021b. 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*, pages 4972–4984, Online. Association for Computational Linguistics.

Ashwin Devaraj, William Sheffield, Byron C Wallace, and Junyi Jessy Li. 2022. Evaluating factuality in text simplification. *arXiv preprint arXiv:2204.07562*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Tianyu Gao, Adam Fisch, and Danqi Chen. 2020. Making pre-trained language models better few-shot learners. *CoRR*, abs/2012.15723.

Jade Goldstein, Mark Kantrowitz, Vibhu Mittal, and Jaime Carbonell. 1999. Summarizing text documents: Sentence selection and evaluation metrics. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 121–128.

Lin Gui, Jia Leng, Gabriele Pergola, Yu Zhou, Ruifeng Xu, and Yulan He. 2019. Neural topic model with reinforcement learning. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3478–3483, Hong Kong, China. Association for Computational Linguistics.

Christian Gulden, Melanie Kirchner, Christina Schüttler, Marc Hinderer, Marvin Kampf, Hans-Ulrich Prokosch, and Dennis Toddenroth. 2019. Extractive summarization of clinical trial descriptions. *International journal of medical informatics*, 129:114–121.

Paul Jaccard. 1912. The distribution of the flora in the alpine zone. 1. *New phytologist*, 11(2):37–50.

Zhengbao Jiang, Frank F. Xu, Jun Araki, and Graham Neubig. 2019. How can we know what language models know? *CoRR*, abs/1911.12543.

Sasikiran Kandula, Dorothy Curtis, and Qing Zeng-Treitler. 2010. A semantic and syntactic text simplification tool for health content. In *AMIA annual symposium proceedings*, volume 2010, page 366. American Medical Informatics Association.

Ilona Kickbusch, Jürgen M. Pelikan, and Franklin Apfel Agis D. Tsouros. 2013. Health literacy : the solid facts.

J Peter Kincaid, Robert P Fishburne Jr, Richard L Rogers, and Brad S Chissom. 1975. Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for navy enlisted personnel. Technical report, Naval Technical Training Command Millington TN Research Branch.

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

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

Yang Liu and Mirella Lapata. 2019. Text summarization with pretrained encoders. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*

(EMNLP-IJCNLP), pages 3730–3740, Hong Kong, China. Association for Computational Linguistics.

Leonardo Campillos Llanos, Dhouha Bouamor, Pierre Zweigenbaum, and Sophie Rosset. 2016. Managing linguistic and terminological variation in a medical dialogue system. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 3167–3173.

Junru Lu, Gabriele Pergola, Lin Gui, Binyang Li, and Yulan He. 2020. CHIME: Cross-passage hierarchical memory network for generative review question answering. pages 2547–2560.

Junru Lu, Xingwei Tan, Gabriele Pergola, Lin Gui, and Yulan He. 2022. Event-centric question answering via contrastive learning and invertible event transformation. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 2377–2389, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Mounica Maddela, Mayank Kulkarni, and Daniel Preotiuc-Pietro. 2022. EntSUM: A data set for entity-centric extractive summarization. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3355–3366, Dublin, Ireland. Association for Computational Linguistics.

Rada Mihalcea and Paul Tarau. 2004. TextRank: Bringing order into text. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 404–411, Barcelona, Spain. Association for Computational Linguistics.

Rashmi Mishra, Jiantao Bian, Marcelo Fiszman, Charlene R Weir, Siddhartha Jonnalagadda, Javed Mostafa, and Guilherme Del Fiol. 2014. Text summarization in the biomedical domain: a systematic review of recent research. *Journal of biomedical informatics*, 52:457–467.

Shashi Narayan, Shay B Cohen, and Mirella Lapata. 2018a. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. *arXiv preprint arXiv:1808.08745*.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018b. Ranking sentences for extractive summarization with reinforcement learning. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1747–1759, New Orleans, Louisiana. Association for Computational Linguistics.

Alok Ranjan Pal and Diganta Saha. 2014. An approach to automatic text summarization using wordnet. In *2014 IEEE International Advance Computing Conference (IACC)*, pages 1169–1173.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Gabriele Pergola, Lin Gui, and Yulan He. 2021a. A disentangled adversarial neural topic model for separating opinions from plots in user reviews. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2870–2883, Online. Association for Computational Linguistics.

Gabriele Pergola, Elena Kochkina, Lin Gui, Maria Liakata, and Yulan He. 2021b. Boosting low-resource biomedical QA via entity-aware masking strategies. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1977–1985, Online. Association for Computational Linguistics.

Fabio Petroni, Tim Rocktäschel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, Alexander H Miller, and Sebastian Riedel. 2019. Language models as knowledge bases? *arXiv preprint arXiv:1909.01066*.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. 2018. Improving language understanding by generative pre-training.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *ArXiv*, abs/1910.01108.

Timo Schick and Hinrich Schütze. 2020. Exploiting cloze questions for few-shot text classification and natural language inference. *CoRR*, abs/2001.07676.

Thibault Sellam, Dipanjan Das, and Ankur P Parikh. 2020. Bleurt: Learning robust metrics for text generation. *arXiv preprint arXiv:2004.04696*.

RJ Senter and Edgar A Smith. 1967. Automated readability index. Technical report, Cincinnati Univ OH.

Ronald Seoh, Ian Birle, Mrinal Tak, Haw-Shiuan Chang, Brian Pinette, and Alfred Hough. 2021. Open aspect target sentiment classification with natural language prompts. *arXiv preprint arXiv:2109.03685*.

Matthew Shardlow and Fernando Alva-Manchego. 2022. Simple tico-19: A dataset for joint translation and simplification of covid-19 texts. In *Proceedings of the 13th Language Resources and Evaluation Conference, Marseille, France, June. European Language Resources Association*.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages

1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Kristine Sørensen, Jürgen M Pelikan, Florian Röthlin, Kristin Ganahl, Zofia Slonska, Gerardine Doyle, James Fullam, Barbara Kondilis, Demosthenes Agrafiotis, Ellen Uiters, et al. 2015. Health literacy in europe: comparative results of the european health literacy survey (hls-eu). *European journal of public health*, 25(6):1053–1058.

Renliang Sun, Hanqi Jin, and Xiaojun Wan. 2021. Document-level text simplification: Dataset, criteria and baseline. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7997–8013, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Zhaoyue Sun, Jiazheng Li, Gabriele Pergola, Byron Wallace, Bino John, Nigel Greene, Joseph Kim, and Yulan He. 2022. PHEE: A dataset for pharmacovigilance event extraction from text. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5571–5587, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed Chi, Quoc Le, and Denny Zhou. 2022. Chain of thought prompting elicits reasoning in large language models. *arXiv preprint arXiv:2201.11903*.

Kristian Woodsend and Mirella Lapata. 2011. Learning to simplify sentences with quasi-synchronous grammar and integer programming. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 409–420, Edinburgh, Scotland, UK. Association for Computational Linguistics.

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

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

Michihiro Yasunaga, Jure Leskovec, and Percy Liang. 2022. Linkbert: Pretraining language models with document links. *arXiv preprint arXiv:2203.15827*.

Hongyi Yuan, Zheng Yuan, Ruyi Gan, Jiaxing Zhang, Yutao Xie, and Sheng Yu. 2022. Biobart: Pretraining and evaluation of a biomedical generative language model. *arXiv preprint arXiv:2204.03905*.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.

Ming Zhong, Pengfei Liu, Yiran Chen, Danqing Wang, Xipeng Qiu, and Xuanjing Huang. 2020. Extractive summarization as text matching. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6197–6208, Online. Association for Computational Linguistics.

Lixing Zhu, Zheng Fang, Gabriele Pergola, Robert Procter, and Yulan He. 2022. Disentangled learning of stance and aspect topics for vaccine attitude detection in social media. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1566–1580, Seattle, United States. Association for Computational Linguistics.

Zhemin Zhu, Delphine Bernhard, and Iryna Gurevych. 2010. A monolingual tree-based translation model for sentence simplification. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 1353–1361, Beijing, China. Coling 2010 Organizing Committee.

A Experimental Setup

A.1 Hyperparameters

For the summarization stage, we adopt NLTK[6] for the building of reference summary dataset, and fine-tune a distilbert-base-uncased-finetuned-sst-2-english[7] PLM as the classifier. The chosen PLM is a distilbert-base-uncased(Sanh et al., 2019) checkpoint additionally fine-tuned on SST-2 dataset(Socher et al., 2013), which is a sentiment binary classification corpus. The hidden size of the checkpoint is 768 and the corresponding vocabulary size is 30,522. The random seed is 42. The batch size is set to 16 and the accumulation steps is set to 1 on 2 quadro_rtx_6000 GPUs. The optimizer is BertAdam[8] with $\beta 1 = 0.9, \beta 2 = 0.999$, and ϵ=1e-6. The weight of decay is 0.01. The learning rate is 2e-5 without warmup. It takes 0.5 hour in total to fine-tune the checkpoint on the training set, and predict over development and testing sets.

And for the simplification stage, except for possible replacement of backbone encoder-decoder PLM, we adopt exact same settings with the SOTA baseline (Devaraj et al., 2021a), including training strategy and sampling method during the predictive generation. It takes less than 20 mins to fine-tune the PLM, while requires 2 hours to generate simplified text over entire testing set on same GPUs.

---

[6]https://www.nltk.org/
[7]https://huggingface.co/distilbert-base-uncased-finetuned-sst-2-english
[8]https://github.com/google-research/bert/blob/master/optimization.py

A.2 Annotation Schema

To overcome the aforementioned limitations on evaluation metrics, we followed a previous approach to ask our annotators give numerical scores for each instances (Alva-Manchego et al., 2021). Considering the requirement on simplification task and feature of text in medical domains (Devaraj et al., 2022), we designed our numerical scores into three aspects: Simplicity, Fluency and Factuality. Annotator can select one numerical score under each aspect, which include three options 0,1 and 2. Higher score stands for annotator consider the paragraph level performance under that aspect is excellent, vice versa. In here, we provide detail explanation of each aspect.

Simplicity aspect considers how simple that text is to read. This category assess the generated text by annotator's impression of simplicity, in terms of length of the texts and use of vocabulary. A good simplified text is expected to omit unnecessary numerical descriptions and explain jargons that are hard to be understood by layman readers.

Fluency aspect considers the how fluent the text is. That is, to assess the simplified text by annotator's impression on connectivity and fluency. A good simplified paragraph should consider the fluency among sentences, such as use of conjunction words or adversative words for sentences. This category also includes the evaluation on overall grammar correctness of each sentences, and penalty on duplicate sentences generated by the model.

Factuality considers how consistent is the simplified text with the original text. This category requires annotators to assess the generated text by compare the facts that mentioned from the original text and those included in the generated text. A good simplified text should includes all the important information appears in the original text. Any paraphrase on the simplified text that lead to different meaning and against the original texts, or any omits on important information should consider to give penalize under this category.

B Robustness of NapSS

We finetune our NapSS model with another two random seeds 123 and 2023. The results of three experiments in 4 share high similarity, confirming the robustness of our proposed pipeline.

Models	Readability		Lexical Similarity				Simplification	Semantic Similarity	Comprehensive
	FK	ARI	Rouge-1	Rouge-2	Rouge-L	BLEU	SARI	BertScore	BLEURT
NapSS (seed=42)	10.97	14.27	48.05	19.94	44.76	12.3	**40.37**	**25.73**	-0.155
NapSS (seed=123)	10.89	14.17	**48.38**	**20.24**	**45.11**	**12.5**	40.36	25.67	-0.149
NapSS (seed=2023)	**10.85**	**14.09**	48.29	20.09	45.02	12.4	40.31	25.60	**-0.148**

Table 4: Robustness checking of our NapSS.

C Summarizer Results

We fine-tuned two different bert-based classifiers, the aforementioned distillbert one, and another BioLinkBERT-base[9], which is a bert-based model pretrained on PubMed abstracts concerning citation links(Yasunaga et al., 2022). Although the BioLink backbone was pretrained on medical corpus, the general Distillbert fine-tuned on similar binary classification dataset performed better.

Models	Accuracy	F1
BioLinkBERT-base	61.91	67.04
Distilbert-base-uncased-finetuned-sst-2-english	62.50	68.91

Table 5: Performance on the constructed testing set.

---

[9] https://huggingface.co/michiyasunaga/BioLinkBERT-base