

# A Transformer Seq2Seq Model with Fast Fourier Transform Layers for Rephrasing and Simplifying Complex Arabic Text

Abdullah Alshantqi<sup>1</sup>, Ahmad Alkhodre<sup>2</sup>, Abdallah Namoun<sup>3</sup>, Sami Albouq<sup>4</sup>,  
Emad Nabil<sup>5</sup>

Faculty of Computer and Information Systems, Islamic University of Madinah  
Madinah 42351, Saudi Arabia<sup>1,2,3,4</sup>  
Faculty of Computers and Artificial Intelligence  
Cairo University, Giza 12613, Egypt<sup>5</sup>

**Abstract**—Text simplification is a fundamental unsolved problem for Natural Language Understanding (NLU) models, which is deemed a hard-to-solve task. Recently, this hard task has aimed to simplify texts with complex linguistic structures and improve their readability, not only for human readers but also for boosting the performance of many natural language processing (NLP) applications. Towards tackling this hard task for the low-resource Arabic NLP, this paper presents a text split-and-rephrase strategy for simplifying complex texts, which depends principally on a sequence-to-sequence Transformer-based architecture (which we call TSimAr). For evaluation, we created a new benchmarking corpus for Arabic text simplification (so-called ATSC) containing 500 articles besides their corresponding simplifications. Through our automatic and manual analyses, experimental results report that our TSimAr evidently outperforms all the publicly accessible state-of-the-art text-to-text generation models for the Arabic language as it achieved the best score on SARI, BLEU, and METEOR metrics of about 0.73, 0.65, and 0.68, respectively.

**Keywords**—Text simplification; sequence-to-sequence; split-and-rephrase; natural language understanding; NLP; TSimAr; ATSC

## I. INTRODUCTION

Texts with complex linguistic structures often pose a difficulty in interpreting and understanding the indented meanings, particularly the meanings between the lines. Such difficulty is not only encountered by human readers but also by intelligent applications that demand text comprehension at some point. Consequently, text simplification methodologies have come to help various readers (mainly readers with low-literacy skills [1], such as children or non-native readers) as well as to boost the performance of many natural language processing (NLP) applications (e.g., automated text parsing [2], summarizations [3], and translations).

Given a linguistically complex text as input, automated text simplifications (ATS) work around generating candidate texts as output that are essentially uncomplicated in structure and easy to understand without losing the purposed meaning [4]. In broad, the relatively common two steps involved in many ATS approaches are 1) splitting complex texts into simple sentences [5], [6] and 2) text rephrasing [7], [8], [9], [10], [11] using more straightforward common words (well-known in some approaches as lexical paraphrasing). In this paper, we

focus on these two steps and introduce a novel text split-and-rephrase solution for Arabic ATS.

Unlike the highly supported Indo-European languages (such as English), the recent ATS literature [4], [12], [13], [14] indicates that quite a few works are dedicated to supporting the Arabic language. To our knowledge, a few text splitting (e.g. [5]) and/or text-to-text rephrasing (e.g. [15], [16], [17], [18], [19]) techniques exist in Arabic NLP literature for tasks not related to simplification problems. Thus, in this paper, we seek to combine these techniques to support Arabic ATS in general. To illustrate the originality of our proposal, in the next section, we review in some detail these techniques besides exploring the existing non-Arabic split-and-rephrase models [7], [20], [10], [9]. Until then, we briefly outline the originality and main contributions of this paper as follows:

- We introduce a text split-and-rephrase strategy for simplifying complex Arabic texts, which depends principally on a sequence-to-sequence Transformer-based architecture. In the splitting part, we integrate our suggested solution with a punctuation detector for text segmentation (PDTs) built on top of a pre-trained multilingual masked-language model (mBERT). This PDTs attempts to generate the shortest set of simple independent-clause sentences from a given lengthy complex text. While in the rephrasing part, we propose a modified attention-free Transformer model, depending on a fast Fourier-Transform (FNet-based), which rephrases the concatenated simple sentences into a more readable version. The significant (original) work introduced in this paper focuses on the latter part.
- We create a new Arabic corpus for benchmarking text simplification approaches and make it publicly available<sup>1</sup>. Besides, we make the details of our experimental evaluations and implementations (i.e., including codes and scripts) publicly accessible<sup>2</sup> to the interested researchers for replicating our experiments.

The remainder of this paper structurally unfolds as follows. First, we present the closely related work and the state-of-the-art NLP pre-trained models, including a brief review of gaps in the literature concerning the Arabic text simplification prob-

<sup>1</sup><https://github.com/AMahfodh/ArSummarizer/tree/main/TSimAr/resources>

<sup>2</sup><https://github.com/AMahfodh/ArSummarizer/tree/main/TSimAr>

lem. Next, we introduce our text split-and-rephrase approach and discuss the subsequent experimental analysis and results. Concluding the paper and outlining the potential forthcoming avenues of investigation are presented afterward.

## II. REVIEW OF RELATED LITERATURE

This section provides a synopsis of Text Simplification (TS) methods, broadly categorized into extractive and abstractive approaches. In principle, text simplification approaches (illustrated in Figure 1) are abstractly analogous to text summarization approaches, as they both strive to simplify texts. However, the dissimilarity between them is fundamental. Text summarization attempts to shorten text without losing the key meanings, while simplification attempts to improve text readability by reducing linguistic complexity constrained by preserving the key meanings. Before delving into TS approaches, we position the novelty of our contribution in the context of related literature by stating the following:

- no split-and-rephrase model has been suggested yet for simplifying complex Arabic texts to the extent of the authors' knowledge; and
- for text-to-text rephrasing, we explore, for the first time, the effect of the modified attention-free Transformer model [21](i.e., depends on a fast Fourier-Transform) for the Arabic language.

### A. Extractive Approaches

The core idea of the extractive approach is to extract the main sentences of the text through text summarization. In other words, the simplification here is performed using summarization. However, summarization does not necessarily lead to simplification. Therefore, this approach is not recommended. One of the examples of text simplification through summarization is the TF-IDF [14]. Preprocessing is a normal requirement of this approach. The preprocessing includes converting text into lower-case, removing punctuation, special characters, and stop words, and stemming to return complex words to their language base.

### B. Abstractive Approaches

The previous approach is not a real simplification, it is just a summarization, but in the abstractive approach, the output text is a real simplified text. In this approach, there are two main categories of simplifications. The first is Lexical Simplification (LS), and the second is Text Generation (TG).

1) *Lexical simplification*: The simplification in the LS category is performed through the replacement of complex/hard words with simple/easy words. Therefore, it is called lexical simplification. LS algorithms work at the sentence level. The structure of sentences is not changed, and the grammar is simplified. Only word replacement is included in the simplification. As so, this type of simplification is not effective enough. The hard words may be replaced with easy ones, but the sentence structure and grammar may still be hard to understand. Examples of lexical simplification are the following.

- Rules-based LS
- Parallel corpora extracted-rules LS
- Word embedding LS

- Pre-trained language models LS

Rule-based LS [22], [23] depends on a linguistic database, like WordNet, to get the simplest synonym of a word, which can be based on its frequency or its length. In parallel corpora, extracted-rules TS [24], [25], [26], the rules are extracted automatically from a parallel aligned corpus. While Word Embedding TS [27], [28], [29] has the advantage that there is no need for lexical resources. The appearance of pre-trained models had a huge benefit for all NLP tasks; one of these tasks is text simplification. Some systems use pre-trained models like BERT to find and generate easy words for complex ones. Therefore, the class of pre-trained language models TS systems [30], [31] has proved its effectiveness compared to other techniques.

2) *Text generation*: The second category of the abstractive approach is Text Generation (TG). In TG, a new simplified text is generated. The new text may have a different structure or a different number of sentences. TG's approach includes sentence splitting, text addition, and deletion. TG can only be considered a true simplification, while the previous methods can be seen as good trials for the simplification, but not real simplification because real simplification means digestion of the original text and generating a new simplified one with new simple words, structures, and grammar. The recent text generation-based TS is data-driven, considering the advantage of the complex structures in data. The text generation techniques can be classified as follows:

- Syntactic simplification
- Statistical Machine Translation (SMT)
- Deep Learning Techniques

In the next paragraphs, we explain each approach.

a) *Syntactic simplification*: In syntactic simplification, the hard/complex words are replaced by easy/simple ones, and the grammatically complex sentences are identified and rewritten in simple sentences. The process of simplification includes splitting long sentences, changing passive sentences to active ones, and resolving ambiguities. Examples of syntactic simplification research are mentioned in [32], [33].

In [34], the authors proposed a model called: TriS: Their approach was to break down a long statement into numerous shorter ones. When a sentence is written in the subject-verb-object order, it is called simple (SVO). A dataset of 854 sentences taken from the New York Times and Wikipedia, manual simplification is performed for the evaluation. The authors assess 100 unseen sentences and compare them to Heilman and Smith's rule-based method [35]. Their approach gets higher ROUGE [36] and Flesch-Kincaid Grade Level scores. Another method is grammar induction, where simplification is considered by converting the tree1 to tree2 problem, where tree1 is for the source, and tree2 is for the target. The process includes extracting tree transformation rules using a corpus, then learning how to select the adequate rule(s) to be applied to simplify unseen sentences. In [37], authors modeled the syntactic and lexical simplification using tree transduction rules. The evaluation of their proposal was performed using the Simple Wikipedia corpus and it showed good results. The authors highlighted the need for a mechanism to eliminate useless transformation rules.

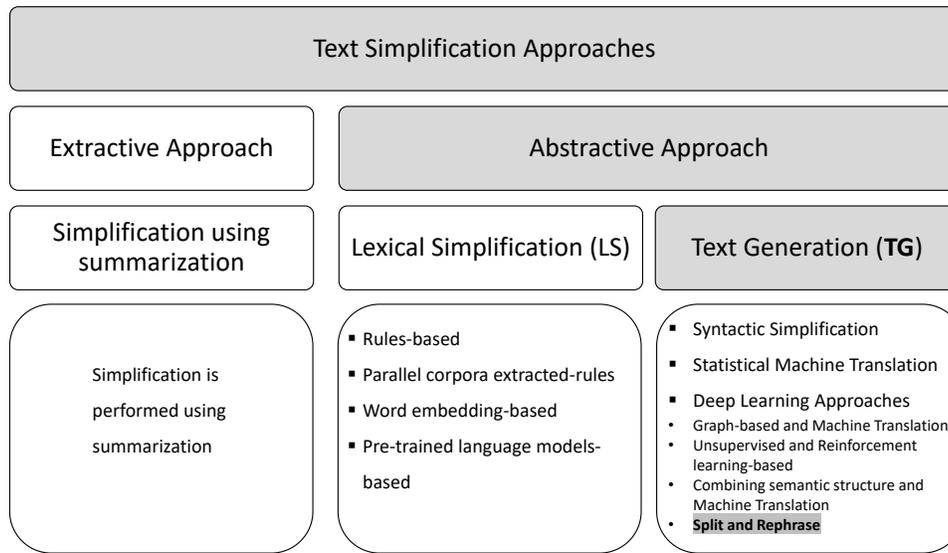


Fig. 1. Text simplification approaches

b) Statistical machine translation (SMT): Machine translation (MT) is the process of translating a text written in language A (source) to language B (target), where the two languages are different. Due to the massive amount of available data nowadays, MT has achieved many success stories. MT is applied successfully to the TS by considering that language B (target) is a simplified version of language A (source), where both represent the same language. Now the problem of simplification can be seen as the generation of monolingual text-to-text or monolingual translation. Some research used phrase-based statistical MT and applied it to TS. The task obviously is simpler than the source, and the target languages are the same; only the target is simpler than the source. Examples of SMT used for TS includes [38], [39], [40], [41], [42].

c) Deep learning techniques: In the era of big data, powerful computers, and GPUs, deep learning took the lead in AI, especially data-driven AI. Deep learning proved to be effective when used with SMT, where RNN Encoder-Decoder is used in MT [43]. This motivated researchers to employ DL in TS using the monolingual translation approach. In [44], the researchers successfully used RNN-based Neural Machine Translation (NMT) for TS; other authors in [45] used LSTM Encoder-Decoder in the simplification process. The authors in [46] developed a model called R-PBMT, Phrase-Based Machine Translation, augmented with a re-Ranking heuristic based on dissimilarity. The model is trained and tested using the PWKP dataset; they compare their work with three models, Word-Substitution Baseline models, that replace a word in a sentence with synonyms retrieved from WordNet. In another research [47], the authors performed four rewriting operations, replacing, splitting, reordering, and deletion; their work is also depending on DL. In [48], the authors proposed a system that is based on quasi-synchronous grammar. Results showed the general superiority of their model using the human evaluation and the automatic evaluation using the metrics BLEU and Flesch-Kincaid grade level. There exist many other DL-based approaches for TS. These approaches include graph-based

approaches [49], reinforcement learning-based TS [50], NMT [51], combining semantic structure and NMT [52], phrase-based unsupervised TS [53], unsupervised neural TS [54], and split-and-rephrasing techniques [8], [9], [7], [10], [20]. In this paper, we conceptually consider the latter technique (i.e., split-and-rephrasing technique) for the Arabic language for the first time. In addition, Table I summarizes the recent existing approaches which indicate the originality of our proposal.

### III. METHODOLOGY

Figure 2 presents an overview of our proposed TSimAr in five steps with illustrative input-output examples. In principle, we integrate a punctuation detector for text segmentation (PDTs [5], see step 3) with a modified attention-free Transformer architecture for rephrasing and simplifying complex Arabic text, step 4. The former (i.e., PDTs) attempts to split a given input text into the shortest set of simple independent-clause sentences. The latter (i.e. the focus of this paper) aims to rephrase the concatenated simple sentences to generate a more readable version. For example, given a textual document  $X$  containing complex sentences, TSimAr attempts to break it down into ( $Y$ ) uncomplicated sentences with rephrasing, such that  $Y \leftarrow TSimAr(X)$  and  $Y = \{y_1, y_2, \dots, y_{|n|}\}$ , where  $n$  is the number of generated simple sentences.

As usual with most NLP applications, TSimAr is set off with straightforward text preprocessing (see step 2), which includes noise/diacritizations removal and soft normalization. This preprocessing step cleans the input texts without breaking sentence structures, and more importantly, it preserves the overall meaning to an extent. For implementing this step, we consider two Python-based toolkits: NLTK<sup>3</sup> and CAMEL<sup>4</sup>. Once the preprocessing step is performed, TSimAr segments and then rephrases the cleaned input text, shown in steps 3 and 4 of Figure 2. In the following subsections, we focus on these

<sup>3</sup>NLTK Tool: <https://www.nltk.org/>

<sup>4</sup>CAMEL is an Arabic natural language processing tool: <https://camel-tools.readthedocs.io/>

TABLE I. A BRIEF SUMMARY OF REVIEWED TEXT SIMPLIFICATION APPROACHES THAT APPLY SPLIT-AND-REPHRASING TECHNIQUES

Approach	Language	Split Technique	Rephrase Technique	Transformer Architecture
[8] Context-Preserving Text Simplification	English	Defining a 35 hand-crafted transformation rules	Semantic hierarchy of minimal propositions	✗
[9] Fact-Aware Sentence Split and Rephrase with Permutation Invariant Training	English	Training a CNN model for predicting sentence split	Seq2seq Model	✗
[7] Controllable Text Simplification with Explicit Paraphrasing	English	DisSim: structural simplification tool consisting of 35 hand-crafted grammar rules	Transformer seq2seq Model	✓
[10] A Memory-Based Sentence Split and Rephrase Model	English	Encoder and Decoder transformer model connected by a memory gate layer		✓
[20] Hierarchical Generation for Sentence Simplification	English	A semantic separator layer at decoder model	LSTM Seq2seq based Model	✗
our TSimAr	Arabic	PDTS [5] (built upon mBERT model)	FNet Text-to-Text based model	✓

two main steps (i.e. the segmentation and rephrase steps) in a little more detail.

### A. Text Segmentation

We base our proposed TSimAr on top of PDTS [5] (i.e. an Arabic text splitting tool that employs a pre-trained multilingual BERT [55] model for detecting missing punctuations) for segmenting input texts into a set of potentially independent-clauses. More in detail, PDTS queries mBERT<sup>5</sup>  $|X|$  times to predict proper punctuations between words on which they can be used as text split delimiters:

$$p(pu_i^m | t_i^m; pun, \theta) = PDTS \circ mBERT(X'_i), \quad (1)$$

$$X'_i \leftarrow insertMaskToken(X, i), \forall i \in X \quad (2)$$

where  $pu_i^m$  represents the valid mBERT's output for a given white-space at index  $i$ ;  $t_i^m$  is the actual mBERT's output;  $pun$  and  $\theta$  are model parameters set by the user to filter out  $t_i^m$ ; and  $X'_i$  is the input  $X$  with the inserted  $[MASK]$  token at index  $i$ . PDTS then validates the predicted set of punctuations  $pu^m \forall i \in \{1, 2, \dots, |X|\}$  using four generic linguistic rules in a greedy-like strategy. In this paper, we have set  $pun$  with only the main splitting punctuations, including full-stop, comma, semicolon, and colon.

### B. Rephrase Generation

Motivated by Transformers-based encoder-decoder architecture [56] that has achieved outstanding improvements in complicated NLP tasks, we consider one of its optimized sequence-to-sequence models. In particular, we utilize (FNet) an efficient version that substitutes complex self-attention layers with linear Fourier Transforms-based layers, introduced originally in [21]. Here, FNet is efficiently lighter and much faster than the standard Transformer model with complicated attention layers. Besides, it closely matches the performance of the standard Transformer model. Considering the right part of Figure 2, it describes the core part of standard Transformer architecture that is modified by adding fast Fourier Transforms layers. Broadly speaking, *Transformer* blocks are stacked with a size of  $N_x$ , where each block consists of a two residual

gateless layers that adds additional weight matrix with *Skip Connection*, Eq. (3):

$$y := \epsilon f(x) + x \quad (3)$$

where  $\epsilon$  is the regularization parameter. In the standard Transformer architecture, the multi-head attention (i.e. concatenate a number of self-attention layers) allows to learn the structural and morphological correlation between different input-tokens impressively, using Eq. (4):

$$attention(Q_i, K_i, V_i) = \sigma \left( \frac{Q_i K_i^T}{\sqrt{d_k}} \right) V_i, \quad \text{for } i = 1, \dots, h \text{ (head)}. \quad (4)$$

where **query**, **key**, and **value** vectors are computed as  $Q_i = W_q^i x$ ,  $K_i = W_k^i x$ ,  $V_i = W_v^i x$ ; and  $\sigma$  is the softmax activation function. Nevertheless, it is memory intensive and has an exponential (quadratic) time complexity concerning the size of the input sequence [56]. Thus, to avoid such scalability issues, we employ fast Fourier Transform layers as an alternative to attention sublayers, expressed in Eq. (5):

$$W_{ij} = \left( \frac{e^{-i \frac{2\pi}{N} i \times j}}{\sqrt{N}} \right) \quad (5)$$

where  $i, j = 0, \dots, N - 1$ . Recent experiments [21] demonstrated that the Fourier Transform-based model (so-called FNet) can significantly reduce the training time and space complexity while providing an excellent performance that is exceptionally comparable to the performance offered by the standard Transformer-based encoder-decoder model. The architecture of our implemented FNet sequence-to-sequence model for rephrasing Arabic texts is presented explicitly in Figure 3.

## IV. EXPERIMENTS AND RESULTS

We conduct experiments to assess the performance of TSimAr and analogize its rephrasing part with the existing Arabic pre-trained text-to-text generation models. We introduce ATSC (a new Arabic corpus for text simplification) and use it in our evaluation protocols. As a quality benchmark of generated simplifications, we report various automatic text

<sup>5</sup><https://github.com/google-research/bert/blob/master/multilingual.md>

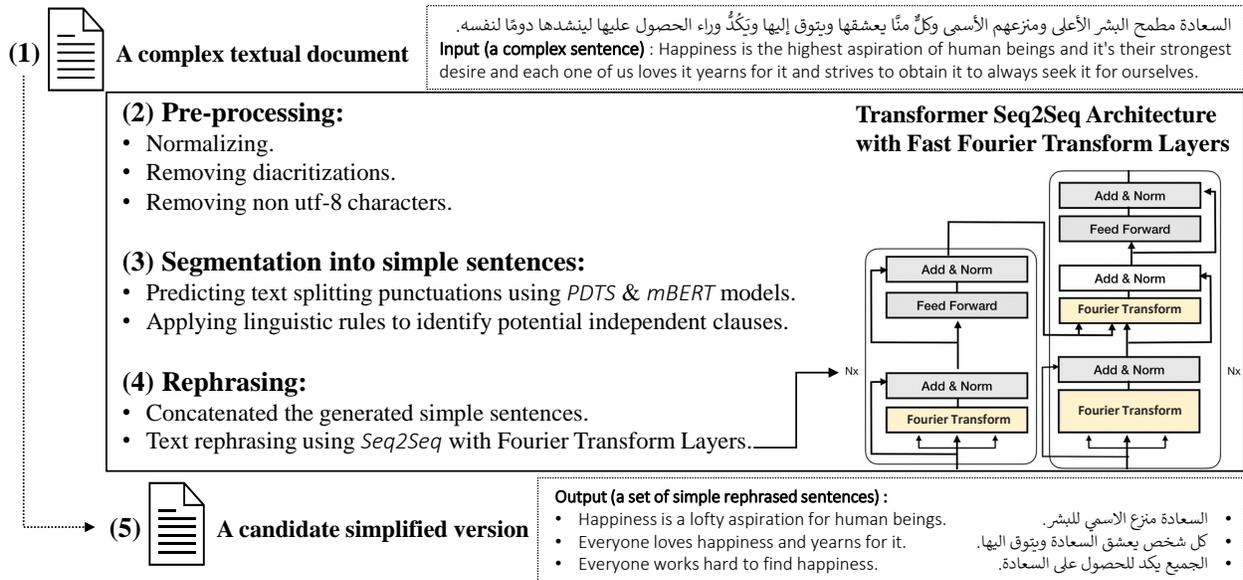


Fig. 2. The overview flow of the proposed TSimAr with an illustrative input-output example

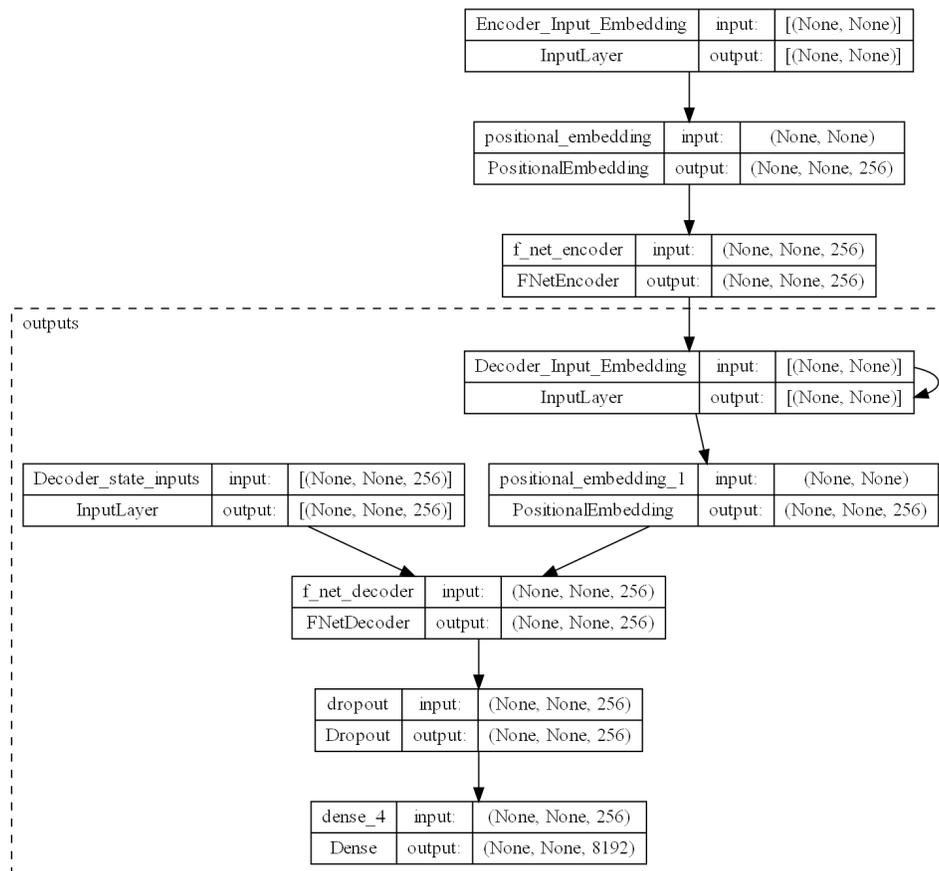


Fig. 3. The architecture of our implemented FNet sequence-to-sequence model for Arabic texts rephrasing. The total trainable parameters are 11,055,616

matching metrics, including SARI (the primary metric for text simplification and rephrasing), besides presenting the findings from the conducted manual (human-based) assessment.

#### A. Corpus and Experimental Setup

a) *Arabic text simplification corpus (ATSC)*:: To the extent of our knowledge, there is no specific Arabic corpus for

text simplification. Thus, we create a small benchmark corpus containing 500 pairs of a small-to-large complex text (source) and a gold-standard simplified (i.e. splitted and rephrased) text. The gold-standard reference simplifications are written and carefully reviewed by human experts. In a little more detail, our corpus ATSC has been constructed from selective Arabic articles that contain appropriate text to simplify. We collected these articles from different public sources (i.e., Wikipedia, newspapers, and news agencies), which cover various domains, including history, geography, health, education, and technology. For constructing the simplified versions from the collected articles (i.e., form the gold-standard human-based references), we have applied two simplification methods: *syntactic simplification* (i.e. just an extractive text summarization method that drops/selects the key sub-sentences without generating new words) as well as *linguistic simplification* (i.e. almost similar to the abstractive summarization method that attempts to replace complicated words with conceivably simpler synonym words), depending on their contexts and overall meaning. Table II shows some general statistical descriptions of our ATSC.

TABLE II. STATISTICAL DESCRIPTIONS OF OUR EVALUATION CORPUS (ATSC)

	Original complex texts	Simplified Rewrites
# No. Documents	500	500
# No. Sentences	546	1243
# No. Words (distinct)	17069 (6046)	17564 (5479)
# No. Vocabularies (for both complex and simplified texts)	6737	

*b) Baselines::* We compare the text rephrasing part in our TSimAr (i.e. FNet model) against the state-of-the-art pre-trained Arabic monolingual (Arabic-T5-small [16], Arabic-T5 [15], UBC-AraT5 [17]) and general multilingual (MT5-base [19], mBART-large-50 [18]) models for text generation tasks. These text-to-text generation models are architecturally extended from T5 encoder-decoder transformer blocks[57], except mBART that is a multilingual Sequence-to-Sequence model used generally for translation tasks:

*c) Automatic metrics::* Given a source text  $st$  and a gold-reference  $gr$  (i.e. a typical simplification version written by human experts), we evaluate the efficiency of the produced simplification  $y$  (i.e.  $Y \leftarrow TSimAr(st)$ ) using a variety of automatic metrics as follows:

- SARI [58] (*System output against References and against the Input sentence*) is a standard evaluation metric for text simplification, which compares the generated candidate simplifications  $y$  against both (1) the source input  $st$  and (2) the gold-reference  $gr$ . It uses precision and F1 scores of  $n$ -grams ( $n \in 1, 2, 3, 4$ ) to measure the goodness of added, deleted, and preserved tokens by the simplifier model (i.e. TSimAr).
- BLEU [59] (Bilingual Evaluation Understudy) is a popular evaluation metric for text quality, commonly used in machine-translated tasks. It compares  $y$  against  $gr$  only and approximates recall and precision metrics using the best match ( $n$ -gram) length and modified  $n$ -gram precision, respectively.
- METEOR [60] (*Metric for Evaluation for Translation with Explicit Ordering*) is similar to BLEU but replaces the best match ( $n$ -gram) length and modified  $n$ -gram

precision with a weighted F-score metric that depends on unigram mapping.

- TER [61] (Translation Edit Rate) which estimates the number of edits required (e.g., adding, deleting, or shifting a word token) to improve  $y$  as matched with  $gr$ .
- ROUGE [36] (Recall-Oriented Understudy for Gisting Evaluation) gives different ROUGE- $n$  metrics, where  $n$  represents the number of overlapping  $n$ -grams between  $y$  and  $gr$ . It uses the standard statistical metrics (precision, recall, and F-measure) for its measurements. In our experiments, we consider ROUGE-1 (unigram overlapping), ROUGE-2 (bi-grams overlapping), and ROUGE-L (the longest identical subsequence overlapping between  $y$  and  $gr$ ).

Concerning SARI, BLEU, METEOR, and ROUGE, higher scores indicate better quality correlated to rational human judgments. In contrast, a lower TER metric (i.e. lower editing scores) indicates better performance.

*d) Implementation details:* To train and configure the text rephrasing part in our TSimAr (presented in Figure 3), we applied a 50—20—30 random split on our ATSC corpus to create train, dev, and test sets, respectively. Besides, we used the Adam optimization algorithm for training with a learning rate of 0.001. The training loop lasts 5k epochs with a batch size of 64 and a maximum sequence length of 256. Moreover, the text-to-text generation models, considered in this paper as baselines, are publicly available at the Hugging Face<sup>6</sup>, under the model (card) names: ‘google/mt5-base’, ‘facebook/mbart-large-50’, ‘flax-community/arabic-t5-small’, ‘UBC-NLP/AraT5-base-title-generation’, and ‘malmarjeh/t5-arabic-text-summarization’. We have constructed these models using PyTorch<sup>7</sup> framework besides utilizing some NLP toolkits for text preprocessing, including NLTK<sup>8</sup> and CAMEL<sup>9</sup>. All experiments have been conducted using a gaming PC equipped with Intel i9-CPU, 64G-RAM, and a single NVIDIA GeForce RTX3070 GPU.

## B. Performance Evaluation

In Table III, we show the performance of our TSimAr with FNet against different text rephrasing models (i.e. depending on text-to-text generation models) using the validation portion from ATSC. Performance results are also visualized in Figure 4. Besides that, we break down the performance details and simplification quality for one input instance in Table IV. As can be observed, TSimAr evidently outperforms all the existing state of the art text-to-text generation models for the Arabic language. It achieves the best score on all standard metrics (particularly SARI) and gives the second to the best score on TER and ROUGE-1. The last column on the right of Table III shows the execution time in second, visualized in Figure 5. Here, our TSimAr gives foreseen poor-to-ordinary time performance as its FNet architecture is quite heavy (consisting of more than 11M trainable parameters).

In addition, giving insight into the text simplifications produced by the competitors’ models, presented in Table IV

<sup>6</sup><https://huggingface.co/>

<sup>7</sup><https://pytorch.org>

<sup>8</sup><https://www.nltk.org/>

<sup>9</sup><https://camel-tools.readthedocs.io/>

TABLE III. AUTOMATIC EVALUATION RESULTS. THE BEST PERFORMANCE FOUND IS INDICATED BY THE ASTERISK\*

	SARI	BLEU	TER	METEOR	ROUGE			ET. (Sec)
					R-1	R-2	R-L	
Arabic-T5-small [16]	0.23	0.02	0.89	0.11	0.23	0.11	0.22	128 *
Arabic-T5 [15]	0.21	0.01	0.94	0.05	0.14	0.04	0.14	145
MT5-base [19]	0.18	0.00	0.98 *	0.02	0.05	0.01	0.04	213
mBART [18]	0.53	0.40	0.52	0.61	0.72 *	0.53	0.70	3685
UBC-AraT5 [17]	0.21	0.00	0.95	0.05	0.12	0.04	0.11	178
Our TSimAr	0.73 *	0.65 *	0.34	0.68 *	0.71	0.67 *	0.71 *	724

TABLE IV. EXAMPLES OF GENERATED TEXT SIMPLIFICATIONS BY RELATED BASELINE MODELS, INCLUDING OURS. THE INPUT EXAMPLE IS TAKEN RANDOMLY FROM THE VALIDATION SET, AND THE SIMPLIFICATIONS' QUALITY IS ILLUSTRATED USING OUR CONSIDERED AUTOMATIC METRICS

**Input: complex sentences**

ويستدعي التعامل مع الاحتياجات الصحية المتزايدة التعقيد اتباع نهج متعدد القطاعات تُدمج فيه السياسات الهادفة لتعزيز الصحة والوقاية، مع وضع حلول تستجيب للمجتمعات المحلية، وتقديم خدمات صحية تتمحور حول الناس. وتشمل الرعاية الصحية الأولية أيضاً العناصر الأساسية اللازمة لتحسين الأمن الصحي ودرء التهديدات الصحية مثل الأوبئة ومقاومة مضادات الميكروبات، من خلال تدابير مثل المشاركة المجتمعية والتعليم، والوصفات الطبية الرشيدة، ومجموعة أساسية من وظائف الصحة العمومية الأساسية، بما في ذلك الترصد.

Dealing with increasingly complex health needs calls for a multisectoral approach in which health promotion and prevention policies are combined, with community-based solutions and people-centred health services. Primary health care also includes the essential elements needed to improve health security and stave off health threats such as epidemics and antimicrobial resistance, through measures such as community participation and education, rational prescription, and a core set of essential public health functions, including surveillance.

Model	Generated text
Arabic-T5-small	Including vital goals, and making provisions بما في ذلك الاهداف الحيويه ، واجراء احكام
Arabic-T5	"الصحة الصحية" تواجه الاحتياجات الصحية المتزايدة المعقدة
MT5-base	"Healthcare" is facing increasingly complex health needs
MT5-base	It can also كما يمكن <extra_id_0>
mBART-large-50	ويستدعي التعامل مع الاحتياجات الصحية المتزايدة التعقيد اتباع نهج متعدد القطاعات تدمج فيه السياسات الهادفة لتعزيز الصحة والوقاية ، مع وضع حلول تستجيب للمجتمعات المحلية ، وتقديم خدمات صحية تتمحور حول الناس . وتشمل الرعاية الصحية الاولى ايضا العناصر الاساسيه اللازمه لتحسين الامن الصحي ودرء التهديدات الصحيه مثل الاوبئه ومقاومه مضادات الميكروبات ، من خلال تدابير مثل المشاركة المجتمعيه والتعليم ، والوصفات الطبيه الرشيدته ، ومجموعه اساسيه من وظائف الصحة العموميه الاساسيه ، بما في ذلك الترصد . . .
UBC-AraT5	وزارة الصحة: الدعم الصحي هو الحل الأمثل لمواجهة الاحتياجات الصحية
Our model	Ministry of Health: Health support is the best solution to meet health needs للتعامل مع الاحتياجات الصحية المعقدة يجب ان نتبع نهج متعدد القطاعات لتعزيز الصحة . يجب تقديم خدمات صحيه تتمحور حول الناس . العناصر الاساسيه للرعايه الصحيه الاوليه هي المشاركة المجتمعيه ، الوصفات الطبيه الرشيدته والوظائف الاساسيه للصحة العامه . To deal with complex health needs we must take a multi-sectoral approach to health promotion. People-centred health services must be provided. The essential elements of primary health care are community participation, rational prescriptions, and basic functions of public health.

TABLE V. AUTOMATIC EVALUATION RESULTS FOR THE TEXT EXAMPLE DESCRIBED IN TABLE IV

	SARI	BLEU	TER	METEOR	ROUGE		
					R-1	R-2	R-L
Arabic-T5-small [16]	0.27	0.00	0.98	0.01	0.04	0.00	0.04
Arabic-T5 [15]	0.27	0.00	1.00	0.01	0.04	0.00	0.04
MT5-base [19]	0.27	0.00	0.98	0.01	0.05	0.00	0.05
mBART [18]	0.44	0.15	1.10	0.57	0.51	0.28	0.50
UBC-AraT5 [17]	0.27	0.00	1.00	0.01	0.04	0.00	0.04
Our TSimAr	0.90 *	0.79 *	0.19	0.81 *	0.92 *	0.87 *	0.92 *

and Table V, one can observe that mid-to-high automatic metrics results may not necessarily reflect valid candidate

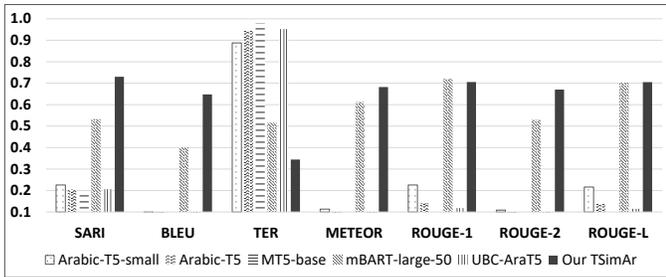


Fig. 4. Performance of TSimAr with FNet against competitors' text rephrasing models (i.e. depending on text-to-text generation models)

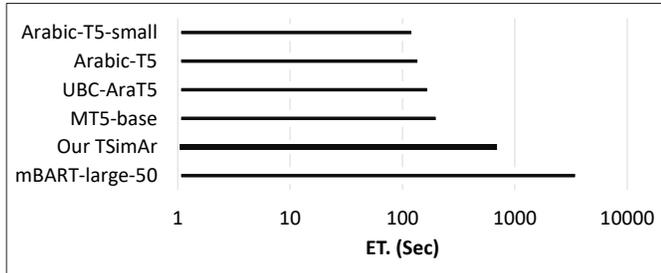


Fig. 5. Performance (w.r.t execution time in second) of TSimAr with FNet against competitors' text rephrasing models

simplification. To clarify more, we observe that mBART [18] often produces outputs almost identical to the input without simplifying or rephrasing, and in turn, it inaccurately archives more than 0.4 SARI score. While, UBC-AraT5 [17] simplifies input text in much better quality, it achieves around 0.27 SARI. Accordingly, it was essential to solidify the evaluation of our proposed TSimAr using manual insight by eliciting human judgments.

### C. Manual Evaluation

To get an additional in-depth evaluation of our TSimAr, we conducted a qualitative analysis by eliciting a humanistic viewpoint on 36 sampled text documents selected randomly from the ATSC validation set. We invited two expert consultants in Arabic linguistics (not authors of this paper) to evaluate these documents (each expert is given 18 documents) on the following three standards using a five-star-point Likert scale (1-5):

- Adequacy (preservation of the source meaning),
- Contextual soundness (quality of rephrased and simplified texts), and
- Grammaticality (to what extent the generated text is free from grammatical errors).

Experts are asked to compare the generated simplifications TSimAr by (i.e. depending on Arabic-T5-small, Arabic-T5, MT5-base, mBART, UBC-AraT5, and our FNet) against the gold-standard references (i.e. text simplification versions written by human experts). With a glancing over into Table VI, the results of our manual evaluation look almost compatible with the automatic evaluation results (shown in Table III) for only the first and the third standards (i.e. *Adequacy* or *Grammaticality*). Nevertheless, *Contextual soundness* standard

TABLE VI. HUMAN EVALUATION RESULTS FOR THE THREE CRITERIA: ADEQUACY, CONTEXTUAL SOUNDNESS, AND GRAMMATICALITY. BASE PRE-TRAINED TEXT-TO-TEXT GENERATION MODELS WITH \* ARE SIGNIFICANTLY DIFFERENT FROM TSIMAR'S BASE MODEL †, DEPENDING ON A TWO-TAILED INDEPENDENT T-TEST, WHERE  $p < .05$ . FOR ARABIC-T5, THE DIFFERENCE IS SIGNIFICANT AT ( $p < .1$ )

Text generation models	A	C	G	Avg.	p-value	t-value
Arabic-T5-small [16]	1.5	2	3	2.17	.010585 *	4.52904
Arabic-T5 [15]	3	3.5	4.5	3.67	.080472 *	1.71791
MT5-base [19]	0	0	1	0.33	.000694 *	9.47046
mBART [18]	5	0	4	3.00	.366645	1.01705
UBC-AraT5 [17]	3.5	4.75	5	4.42	.778051	0.30151
Our TSimAr depending on FNet	4	4.75	5	4.58	†	

reveals the quality differences more precisely, which also confirms that our TSimAr with FNet (indicated by †) can produce a highly competitive performance (see, 4.58 the best average ratings obtained by TSimAr).

Furthermore, the manual experimental results indicated that UBC-AraT5 is a feasible pre-trained text rephrasing model to adopt (i.e. an alternative model to FNet) as it achieves the second highest average score of about 4.42. This indication can also be statistically noticed by its insignificant p-value (i.e. the obtained .78 with UBC-AraT5). In contrast, however, the worst performances observed were with MT5-base and mBART, which unexpectedly gave zero simplification quality. Here, the performance of mBART contradicts the insignificant p-value (i.e. .367) as this heavy model offers an illusive high score in *Adequacy* and *Grammaticality* standards, which is a consequence of generating texts exactly similar to the input texts with no simplification.

### D. Discussion and Potential Threats to Validity

In this section, we discuss the potential threats to the empirical validity of the proposed TSimAr. The main threats may include the creation of our corpus (ATSC) for evaluation as well as the benchmarking against the state-of-the-art text rephrasing models. As mentioned earlier in this section, there is no specific Arabic corpus for text simplification available to date. Therefore, we had to make an effort to (1) collect professionally written corpus from online sources (mainly from newspaper articles) and (2) simplify them precisely (i.e. split with rephrasing) by linguistic experts in Arabic, elaborated in ATSC. One may argue that ATSC is relatively small (containing only 500 pairs of texts), and more importantly, it may be insufficient to train a heavy model like FNet. These thoughts are valid to a reasonable extent. However, recent studies demonstrated that training a language understanding model on a larger corpora/dataset might not necessarily imply improving its performance [62], [63]. Besides, our intention here is not to use ATSC to train a language understating model but rather to use it as a benchmarking corpus for testing the generalization of a pre-trained test-to-text generation model. We make our ATSC available for researchers to exploit in this regard.

Concerning the chosen text rephrasing baseline models, we attempted to counter this concern by using all publicly available Arabic monolingual sequence-to-sequence models (we have encountered only Arabic-T5, Arabic-T5-small, and UBC-AraT5) as well as using the state-of-the-art multilingual models

(i.e. MT5 and mBART). For a fair comparison between these pre-trained models, we confirmed that their large vocabulary contains all the distinct 6737 words extracted from ATSC.

## V. CONCLUSION

Approaching towards breaking down a given complex Arabic text into a simple and meaning-preserving version, we have presented a text split-and-rephrase solution (so-called TSimAr), which depends principally on a sequence-to-sequence Transformer-based architecture. For the splitting, we have integrated TSimAr with a punctuation detector for text segmentation (PDTS) built on top of a pre-trained multilingual masked-language model (mBERT). This PDTS attempts to generate the shortest set of simple independent-clause sentences from a given lengthy complex text. While in the rephrasing phase, we have proposed an attention-free Transformer model, depending on a fast Fourier-Transform (FNet-based), which rephrases the concatenated simple sentences into a more readable version.

In addition, we have created a new corpus (ATSC) to train and evaluate the rephrasing part in our TSimAr. Automated and manual analyses demonstrated that with the support of PDTS, our TSimAr evidently outperforms all the existing state-of-the-art text-to-text generation models for the Arabic language as it achieved the best score on SARI, BLEU, and METEOR metrics. Nevertheless, a trivial limitation noted in TSimAr lies in the execution time compared with competitors' lighter models, such as Arabic-T5-small. Hence, for the generality, we imagine a remarkable extension of this ongoing work in two directions:

- (1) evaluating TSimAr on a comprehensively benchmarking dataset that we plan to create, and
- (2) optimizing our FNet architecture for enhancing its execution performance.

For the latter direction, we will investigate the feasibility of applying a knowledge distillation technique to compress our FNet into a smaller version to help us reduce its space complexity while achieving higher inference speed and accuracy.

## ACKNOWLEDGMENT

The authors extend their appreciation to the Deputyship for Research & Innovation at the Ministry of Education in the Kingdom of Saudi Arabia for funding this research work through project number (19/20).

## REFERENCES

- [1] O. Alonzo, "The use of automatic text simplification to provide reading assistance to deaf and hard-of-hearing individuals in computing fields," *SIGACCESS Access. Comput.*, vol. 3, no. 132, mar 2022. [Online]. Available: <https://doi-org.sdl.idm.oclc.org/10.1145/3523265.3523268>
- [2] T. Hao, X. Li, Y. He, F. L. Wang, and Y. Qu, "Recent progress in leveraging deep learning methods for question answering," *Neural Computing and Applications*, vol. 34, pp. 1–19, 2022.
- [3] A. Alsharqiti, A. Namoun, A. Alsughayyir, A. M. Mashraqi, A. R. Gilal, and S. S. Albouq, "Leveraging distilbert for summarizing arabic text: An extractive dual-stage approach," *IEEE Access*, vol. 9, pp. 135 594–135 607, 2021.
- [4] F. Alva-Manchego, C. Scarton, and L. Specia, "Data-driven sentence simplification: Survey and benchmark," *Computational Linguistics*, vol. 46, no. 1, pp. 135–187, 2020.

- [5] A. M. Alsharqiti, S. Albouq, A. B. Alkhodre, A. Namoun, and E. Nabil, "Employing a multilingual transformer model for segmenting unpunctuated arabic text," *Applied Sciences*, vol. 12, no. 20, 2022. [Online]. Available: <https://www.mdpi.com/2076-3417/12/20/10559>
- [6] C. Niklaus, M. Cetto, A. Freitas, and S. Handschuh, "Transforming complex sentences into a semantic hierarchy," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, July 2019, 57th Annual Meeting of the Association for Computational Linguistics ; Conference date: 28-07-2019 Through 02-08-2019.
- [7] M. Maddela, F. Alva-Manchego, and W. Xu, "Controllable text simplification with explicit paraphrasing," *arXiv preprint arXiv:2010.11004*, 2020.
- [8] C. Niklaus, M. Cetto, A. Freitas, and S. Handschuh, "Context-preserving text simplification," *arXiv preprint arXiv:2105.11178*, 2021.
- [9] Y. Guo, T. Ge, and F. Wei, "Fact-aware sentence split and rephrase with permutation invariant training," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, 2020, pp. 7855–7862.
- [10] X. Fan, Y. Liu, G. Liu, and B. Su, "A memory-based sentence split and rephrase model with multi-task training," in *International Conference on Neural Information Processing*. Springer, 2020, pp. 643–654.
- [11] L. Martin, A. Fan, É. de la Clergerie, A. Bordes, and B. Sagot, "Muss: multilingual unsupervised sentence simplification by mining paraphrases," *arXiv preprint arXiv:2005.00352*, 2020.
- [12] D. Gamal, M. Alfonse, S. M. Jiménez-Zafra, and M. Aref, "Survey of arabic machine translation, methodologies, progress, and challenges," in *2022 2nd International Mobile, Intelligent, and Ubiquitous Computing Conference (MIUCC)*. IEEE, 2022, pp. 378–383.
- [13] M. Zhou, N. Duan, S. Liu, and H.-Y. Shum, "Progress in neural nlp: Modeling, learning, and reasoning," *Engineering*, vol. 6, no. 3, pp. 275–290, 2020. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2095809919304928>
- [14] P. Sikka and V. Mago, "A Survey on Text Simplification," May 2022, arXiv:2008.08612 [cs]. [Online]. Available: <http://arxiv.org/abs/2008.08612>
- [15] F. Community, "Arabic-t5 (small)," 2022. [Online]. Available: <https://huggingface.co/malmarjeh/t5-arabic-text-summarization>
- [16] M. B. Almarjeh, "An arabic abstractive text summarization model," 2022. [Online]. Available: <https://huggingface.co/flax-community/arabic-t5-small>
- [17] E. M. B. Nagoudi, A. Elmadany, and M. Abdul-Mageed, "AraT5: Text-to-text transformers for Arabic language generation," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Dublin, Ireland: Association for Computational Linguistics, may 2022, pp. 628–647. [Online]. Available: <https://aclanthology.org/2022.acl-long.47>
- [18] Y. Tang, C. Tran, X. Li, P.-J. Chen, N. Goyal, V. Chaudhary, J. Gu, and A. Fan, "Multilingual translation with extensible multilingual pretraining and finetuning," *arXiv eprint 2008.00401*, 2020.
- [19] L. Xue, N. Constant, A. Roberts, M. Kale, R. Al-Rfou, A. Siddhant, A. Barua, and C. Raffel, "mt5: A massively multilingual pre-trained text-to-text transformer," *CoRR*, vol. abs/2010.11934, 2020. [Online]. Available: <https://arxiv.org/abs/2010.11934>
- [20] M. Wang, H. Ozaki, Y. Koreeda, and K. Yanai, "Split first and then rephrase: Hierarchical generation for sentence simplification," in *International Conference of the Pacific Association for Computational Linguistics*. Springer, 2019, pp. 15–27.
- [21] I. Eckstein, J. P. Lee-Thorp, J. Ainslie, and S. Ontanon, Eds., *FNet: Mixing Tokens with Fourier Transforms*, 2022. [Online]. Available: <https://aclanthology.org/2022.naacl-main.319.pdf>
- [22] E. Pavlick and C. Callison-Burch, "Simple PPDB: A Paraphrase Database for Simplification," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Berlin, Germany: Association for Computational Linguistics, Aug. 2016, pp. 143–148. [Online]. Available: <https://aclanthology.org/P16-2024>
- [23] M. Maddela and W. Xu, "A Word-Complexity Lexicon and A Neural Readability Ranking Model for Lexical Simplification," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium: Association for

- Computational Linguistics, Oct. 2018, pp. 3749–3760. [Online]. Available: <https://aclanthology.org/D18-1410>
- [24] M. Yatskar, B. Pang, C. Danescu-Niculescu-Mizil, and L. Lee, “For the sake of simplicity: Unsupervised extraction of lexical simplifications from Wikipedia,” in *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Los Angeles, California: Association for Computational Linguistics, Jun. 2010, pp. 365–368. [Online]. Available: <https://aclanthology.org/N10-1056>
- [25] C. Horn, C. Manduca, and D. Kauchak, “Learning a Lexical Simplifier Using Wikipedia,” in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Baltimore, Maryland: Association for Computational Linguistics, Jun. 2014, pp. 458–463. [Online]. Available: <https://aclanthology.org/P14-2075>
- [26] O. Biran, S. Brody, and N. Elhadad, “Putting it Simply: a Context-Aware Approach to Lexical Simplification,” in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. Portland, Oregon, USA: Association for Computational Linguistics, Jun. 2011, pp. 496–501. [Online]. Available: <https://aclanthology.org/P11-2087>
- [27] G. Glavaš and S. Štajner, “Simplifying Lexical Simplification: Do We Need Simplified Corpora?” in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*. Beijing, China: Association for Computational Linguistics, Jul. 2015, pp. 63–68. [Online]. Available: <https://aclanthology.org/P15-2011>
- [28] G. Paetzold and L. Specia, “Lexical Simplification with Neural Ranking,” in *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*. Valencia, Spain: Association for Computational Linguistics, Apr. 2017, pp. 34–40. [Online]. Available: <https://aclanthology.org/E17-2006>
- [29] S. Gooding and E. Kochmar, “Recursive Context-Aware Lexical Simplification,” 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)*. Hong Kong, China: Association for Computational Linguistics, Nov. 2019, pp. 4853–4863. [Online]. Available: <https://aclanthology.org/D19-1491>
- [30] J. Qiang, Y. Li, Y. Zhu, Y. Yuan, and X. Wu, “Lexical Simplification with Pretrained Encoders,” Oct. 2020, arXiv:1907.06226 [cs]. [Online]. Available: <http://arxiv.org/abs/1907.06226>
- [31] W. Zhou, T. Ge, K. Xu, F. Wei, and M. Zhou, “BERT-based Lexical Substitution,” in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, Jul. 2019, pp. 3368–3373. [Online]. Available: <https://aclanthology.org/P19-1328>
- [32] S. Aluísio and C. Gasperin, “Fostering Digital Inclusion and Accessibility: The PorSimples project for Simplification of Portuguese Texts,” in *Proceedings of the NAACL HLT 2010 Young Investigators Workshop on Computational Approaches to Languages of the Americas*. Los Angeles, California: Association for Computational Linguistics, Jun. 2010, pp. 46–53. [Online]. Available: <https://aclanthology.org/W10-1607>
- [33] S. Jonnalagadda and G. Gonzalez, “BioSimplify: an open source sentence simplification engine to improve recall in automatic biomedical information extraction,” *AMIA ... Annual Symposium proceedings / AMIA Symposium. AMIA Symposium*, vol. 2010, pp. 351–5, Nov. 2010.
- [34] N. Bach, Q. Gao, S. Vogel, and A. Waibel, “TriS: A Statistical Sentence Simplifier with Log-linear Models and Margin-based Discriminative Training,” in *Proceedings of 5th International Joint Conference on Natural Language Processing*. Chiang Mai, Thailand: Asian Federation of Natural Language Processing, Nov. 2011, pp. 474–482. [Online]. Available: <https://aclanthology.org/I11-1053>
- [35] M. Heilman and N. A. Smith, “Extracting Simplified Statements for Factual Question Generation.”
- [36] C.-Y. Lin, “ROUGE: A Package for Automatic Evaluation of Summaries,” in *Text Summarization Branches Out*. Barcelona, Spain: Association for Computational Linguistics, Jul. 2004, pp. 74–81. [Online]. Available: <https://aclanthology.org/W04-1013>
- [37] G. H. Paetzold and L. Specia, “Text Simplification as Tree Transduction,” in *Proceedings of the 9th Brazilian Symposium in Information and Human Language Technology*, 2013. [Online]. Available: <https://aclanthology.org/W13-4813>
- [38] M. Shardlow, “A Survey of Automated Text Simplification,” *International Journal of Advanced Computer Science and Applications*, vol. 4, no. 1, Apr. 2014, number: 1 Publisher: The Science and Information (SAI) Organization Limited. [Online]. Available: <https://thesai.org/Publications/ViewPaper?Volume=4&Issue=1&Code=SpecialIssue&SerialNo=9>
- [39] W. Coster and D. Kauchak, “Learning to Simplify Sentences Using Wikipedia,” in *Proceedings of the Workshop on Monolingual Text-To-Text Generation*. Portland, Oregon: Association for Computational Linguistics, Jun. 2011, pp. 1–9. [Online]. Available: <https://aclanthology.org/W11-1601>
- [40] H.-B. Chen, H.-H. Huang, H.-H. Chen, and C.-T. Tan, “A Simplification-Translation-Restoration Framework for Cross-Domain SMT Applications,” in *Proceedings of COLING 2012*. Mumbai, India: The COLING 2012 Organizing Committee, Dec. 2012, pp. 545–560. [Online]. Available: <https://aclanthology.org/C12-1034>
- [41] D. Klaper, S. Ebling, and M. Volk, “Building a German/Simple German Parallel Corpus for Automatic Text Simplification,” in *Proceedings of the Second Workshop on Predicting and Improving Text Readability for Target Reader Populations*. Sofia, Bulgaria: Association for Computational Linguistics, Aug. 2013, pp. 11–19. [Online]. Available: <https://aclanthology.org/W13-2902>
- [42] S. Stymne, J. Tiedemann, C. Hardmeier, and J. Nivre, “Statistical Machine Translation with Readability Constraints,” in *Proceedings of the 19th Nordic Conference of Computational Linguistics (NODALIDA 2013)*. Oslo, Norway: Linköping University Electronic Press, Sweden, May 2013, pp. 375–386. [Online]. Available: <https://aclanthology.org/W13-5634>
- [43] K. Cho, B. van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, “Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation,” in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar: Association for Computational Linguistics, Oct. 2014, pp. 1724–1734. [Online]. Available: <https://aclanthology.org/D14-1179>
- [44] T. Wang, P. Chen, J. Rochford, and J. Qiang, “Text simplification using Neural Machine Translation,” in *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, ser. AAAI’16. Phoenix, Arizona: AAAI Press, Feb. 2016, pp. 4270–4271.
- [45] T. Wang, P. Chen, K. Amaral, and J. Qiang, “An Experimental Study of LSTM Encoder-Decoder Model for Text Simplification,” Sep. 2016, arXiv:1609.03663 [cs]. [Online]. Available: <http://arxiv.org/abs/1609.03663>
- [46] S. Wubben, A. van den Bosch, and E. Krahmer, “Sentence Simplification by Monolingual Machine Translation,” in *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Jeju Island, Korea: Association for Computational Linguistics, Jul. 2012, pp. 1015–1024. [Online]. Available: <https://aclanthology.org/P12-1107>
- [47] Z. Zhu, D. Bernhard, and I. Gurevych, “A Monolingual Tree-based Translation Model for Sentence Simplification,” in *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*. Beijing, China: Coling 2010 Organizing Committee, Aug. 2010, pp. 1353–1361. [Online]. Available: <https://aclanthology.org/C10-1152>
- [48] K. Woodsend and M. Lapata, “Learning to Simplify Sentences with Quasi-Synchronous Grammar and Integer Programming,” in *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*. Edinburgh, Scotland, UK.: Association for Computational Linguistics, Jul. 2011, pp. 409–420. [Online]. Available: <https://aclanthology.org/D11-1038>
- [49] A. See, P. J. Liu, and C. D. Manning, “Get To The Point: Summarization with Pointer-Generator Networks,” Apr. 2017, arXiv:1704.04368 [cs]. [Online]. Available: <http://arxiv.org/abs/1704.04368>
- [50] X. Zhang and M. Lapata, “Sentence Simplification with Deep Reinforcement Learning,” in *Proceedings of the 2017 Conference on*

- Empirical Methods in Natural Language Processing*. Copenhagen, Denmark: Association for Computational Linguistics, Sep. 2017, pp. 584–594. [Online]. Available: <https://aclanthology.org/D17-1062>
- [51] S. Štajner and H. Saggion, “Data-Driven Text Simplification,” in *Proceedings of the 27th International Conference on Computational Linguistics: Tutorial Abstracts*. Santa Fe, New Mexico, USA: Association for Computational Linguistics, Aug. 2018, pp. 19–23. [Online]. Available: <https://aclanthology.org/C18-3005>
- [52] E. Sulem, O. Abend, and A. Rappoport, “Semantic Structural Evaluation for Text Simplification,” Oct. 2018, arXiv:1810.05022 [cs]. [Online]. Available: <http://arxiv.org/abs/1810.05022>
- [53] J. Qiang and X. Wu, “Unsupervised Statistical Text Simplification,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 33, no. 4, pp. 1802–1806, Apr. 2021, conference Name: IEEE Transactions on Knowledge and Data Engineering.
- [54] S. Surya, A. Mishra, A. Laha, P. Jain, and K. Sankaranarayanan, “Unsupervised Neural Text Simplification,” in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, Jul. 2019, pp. 2058–2068. [Online]. Available: <https://aclanthology.org/P19-1198>
- [55] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding,” *arXiv preprint arXiv:1810.04805*, 2018.
- [56] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, “Attention is all you need,” in *Advances in neural information processing systems*, 2017, pp. 5998–6008.
- [57] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, P. J. Liu *et al.*, “Exploring the limits of transfer learning with a unified text-to-text transformer.” *J. Mach. Learn. Res.*, vol. 21, no. 140, pp. 1–67, 2020.
- [58] W. Xu, C. Napoles, E. Pavlick, Q. Chen, and C. Callison-Burch, “Optimizing statistical machine translation for text simplification,” *Transactions of the Association for Computational Linguistics*, vol. 4, pp. 401–415, 2016.
- [59] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, “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.
- [60] S. Banerjee and A. Lavie, “METEOR: An automatic metric for MT evaluation with improved correlation with human judgments,” in *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*. Ann Arbor, Michigan: Association for Computational Linguistics, Jun 2005, pp. 65–72. [Online]. Available: <https://www.aclweb.org/anthology/W05-0909>
- [61] M. Snover, B. Dorr, R. Schwartz, L. Micciulla, and J. Makhoul, “A study of translation edit rate with targeted human annotation,” in *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas: Technical Papers*. Cambridge, Massachusetts, USA: Association for Machine Translation in the Americas, Aug 2006, pp. 223–231. [Online]. Available: <https://aclanthology.org/2006.amta-papers.25>
- [62] G. Inoue, B. Alhafni, N. Baimukan, H. Bouamor, and N. Habash, “The interplay of variant, size, and task type in Arabic pre-trained language models,” in *Proceedings of the Sixth Arabic Natural Language Processing Workshop*. Kyiv, Ukraine (Online): Association for Computational Linguistics, April 2021, pp. 92–104.
- [63] L. Zhang, H. Zhu, S. Brahma, and Y. Li, “Small but mighty: New benchmarks for split and rephrase,” *arXiv preprint arXiv:2009.08560*, 2020.