International Journal of Advanced and Applied Sciences, 12(4) 2025, Pages: 71-78

Contents lists available at Science-Gate

International Journal of Advanced and Applied Sciences

Journal homepage: http://www.science-gate.com/IJAAS.html

A light but efficient switch transformer model for Arabic text simplification

Aeshah Alsughayyir [1, *], Abdullah Alshanqiti [2]

[1]*College of Computer Science and Engineering, Taibah University, Madinah 42353, Saudi Arabia*
[2]*Faculty of Computer and Information Systems, Islamic University of Madinah, Madinah 42351, Saudi Arabia*

ARTICLE INFO

Article history:
Received 14 November 2024
Received in revised form
27 March 2025
Accepted 15 April 2025

Keywords:
Arabic text simplification
Natural language understanding
Low-resource NLP
Transformer model
Text generation

ABSTRACT

Simplifying Arabic text remains a significant challenge in the field of Natural Language Understanding (NLU), making it difficult for current models to perform well. Recent studies have focused on simplifying texts with complex language structures to improve readability for both human users and other Natural Language Processing (NLP) tasks. This study addresses the challenge in the context of low-resource Arabic NLP by introducing a split-and-rephrase approach using a sequence-to-sequence switch transformer model, called ATSimST. Experiments using the ATSC dataset show that ATSimST performs better than existing advanced text generation models for Arabic. The improvements in SARI, BLEU, METEOR, and ROUGE scores demonstrate that ATSimST produces high-quality simplifications that are both semantically accurate and similar to human-written texts. These results confirm the model's effectiveness and highlight its potential to significantly advance Arabic text simplification.

1. Introduction

Text simplification, a crucial area within Natural Language Processing (NLP), aims to make complex text more accessible to a broader audience. This is particularly important for languages like Arabic, which exhibit complex morphological structures and syntactic variations (Alshanqiti et al., 2021). Simplified Arabic text can benefit various applications, including education, accessibility for individuals with cognitive impairments, and cross-lingual information retrieval. While several approaches have been proposed for text simplification (Alshanqiti et al., 2023; Al-Thanyyan and Azmi, 2023), challenges remain in preserving the original meaning while ensuring fluency and grammatical correctness in the simplified output. Texts containing intricate linguistic structures often present challenges for human readers and intelligent applications in accurately interpreting and comprehending the intended meaning, including subtle nuances. This difficulty has spurred the development of text simplification methodologies to aid readers with limited literacy skills, such as children or non-native speakers, and to enhance the performance of various NLP applications like automated text parsing (Hao et al., 2022), summarization (Alshanqiti et al., 2021; Karim et al., 2024), and simplification (Alshanqiti et al., 2023; Al-Thanyyan and Azmi, 2023).

Automated Text Simplification (ATS) systems take complex text as input and generate simplified versions that are easier to understand while retaining the original meaning (Alva-Manchego et al., 2020). Common ATS approaches involve two key steps: (1) decomposing complex sentences into simpler ones (Alshanqiti et al., 2022; Niklaus et al., 2019) and (2) rephrasing the text (Maddela et al., 2020; Niklaus et al., 2021; Guo et al., 2020; Fan et al., 2020; Martin et al., 2020) using more common vocabulary (lexical paraphrasing). This paper focuses on these two steps and extends our previous split-and-rephrase solution (Alshanqiti et al., 2023) for further performance optimization.

Existing research (e.g., Alva-Manchego et al. (2020), Gamal et al. (2022), Zhou et al. (2020), and Alkaldi and Inkpen (2023)) reveals a need for more ATS resources for Arabic compared to well-resourced Indo-European languages like English. While some Arabic NLP techniques exist (Alshanqiti et al., 2022) for text splitting and text-to-text rephrasing (Almarjeh, 2022; Nagoudi et al., 2022; Tang et al., 2020; Xue et al., 2020), they are often applied to tasks unrelated to simplification. This paper aims to bridge this gap by combining these techniques to facilitate Arabic ATS. The following

* Corresponding Author.
Email Address: asughayyir@taibahu.edu.sa (A. Alsughayyir)
https://doi.org/10.21833/ijaas.2025.04.009
Corresponding author's ORCID profile:
https://orcid.org/0000-0003-3710-7103

section provides a detailed review of these techniques in Arabic NLP and explores existing non-Arabic split-and-rephrase models (Maddela et al., 2020; Wang et al., 2019; Fan et al., 2020; Guo et al., 2020) to highlight the originality of our proposed approach.

Alshanqiti et al. (2023) proposed (TSimAr) a text split-and-rephrase strategy, depending on a modified attention-free Transformer model. In this paper, however, we study the same approach but with optimizing attention layers using the Switch Transformer layer for improving the overall computational performance (we call it ATSimST, standing for Arabic Text Simplification using Switch Transformer). ATSimST leverages the efficiency of the Switch Transformer architecture, combined with a preprocessing step for text segmentation, to generate high-quality simplified Arabic text. We evaluate ATSimST against TSimAr besides several state-of-the-art models on the ATSC benchmark dataset (github.com/AMahfodh/ArSummarizer/tree/main/TSimAr/resources), demonstrating its performance across various evaluation metrics. The following sections review related work focusing on current state-of-the-art NLP pre-trained models and then detail (1) the architecture of ATSimST, (2) the experimental setup, and (3) the results of our comparative analysis. Afterward, the paper is concluded, and potential future research avenues are outlined.

2. Related works

This section briefly reviews Arabic Text Simplification methods, divided into extractive lexical simplification and abstractive text generation. While similar to text summarization in aiming to simplify text, TS focuses on reducing linguistic complexity to enhance readability, not just shortening text.

2.1. Extractive lexical simplification

Lexical Simplification (LS) functions by replacing lexically complex or challenging terms with simpler, more accessible equivalents. LS algorithms often operate at the sentence level, maintaining sentential structure and concentrating exclusively on lexical substitution without altering fundamental grammatical frameworks. As a result, the effectiveness of this method may be constrained. Although individual words may be reduced, ongoing difficulties in syntax and grammar may still hinder understanding. Below are illustrative examples of lexical simplification:

- Rules-based LS (Maddela and Xu, 2018; Al-Thanyyan and Azmi, 2023).
- Parallel corpora extracted-rules LS (Horn et al., 2014).
- Word embedding LS (Gooding and Kochmar, 2019).

Rule-based LS methodologies (Maddela and Xu, 2018; Al-Thanyyan and Azmi, 2023) rely on predefined linguistic rules to simplify text. It often utilizes language resources, such as WordNet, to ascertain the most straightforward synonym for a specified phrase, frequently determined by frequency or length. Extracted rules TS techniques (Horn et al., 2014) autonomously generate rules from aligned parallel corpora. Word Embedding TS methods (Gooding and Kochmar, 2019) provide the unique benefit of eliminating the necessity for external lexical resources.

2.2. Abstractive text generation

Text Generation (TG) entails creating wholly original, simplified text that may display varied structures, sentence counts, and vocabulary selections. This method includes procedures like sentence segmentation, text augmentation, and deletion. Although previous methods offer significant insights into simplification, TG arguably represents genuine simplification, as it involves the semantic analysis of the original text followed by the creation of a new, simplified version distinguished by an uncomplicated lexicon, syntax, and grammar. TG techniques can be categorized into: syntactic simplification, statistical machine translation, and Deep Learning Techniques (Stajner and Saggion, 2018; Qiang and Wu, 2021;ˇSurya et al., 2019; Niklaus et al., 2021; Guo et al., 2020; Maddela et al., 2020; Fan et al., 2020; Wang et al., 2019; Alkaldi and Inkpen, 2023). In this paper, we follow the latter technique. Deep Learning Techniques have shown promising results in learning the simplification process directly from data, without the need for explicit rules or parallel corpora. Limited deep learning studies, however, have been suggested for Arabic text simplification (Al-Thanyyan and Azmi, 2023; Alkaldi and Inkpen, 2023; Espinosa-Zaragoza et al., 2023; North et al., 2024; Alshanqiti et al., 2023). For instance, North et al. (2024) presented a multi-task learning framework that jointly performs text simplification and lexical simplification, demonstrating the benefits of this joint approach. Recently, Urakawa et al. (2024) showed a high-fidelity Japanese news simplification corpus of 7,075 sentence pairs. Their evaluation shows better accuracy and easier reading, particularly for non-native readers. Their corpus helps improve the BART model and supports few-shot learning in GPT-3.5. Yuan et al. (2024) introduced SeqDiffuSeq, an encoder-decoder Transformer-based continuous diffusion model for sequence-to-sequence text generation. Leveraging self-conditioning and an adaptive noise schedule for improved performance, SeqDiffuSeq demonstrates superior generation quality and inference speed. However, their experiments in text simplification show poor performance (ranged from 28-37 BLEU score).

In this paper, we consider the second category using deep learning techniques (i.e., abstractive text generation using the split-and-rephrasing technique) for the Arabic language. We extend our previous work (Alshanqiti et al., 2023) and suggest ATSimST (a transformer-based architecture with a switch attention mechanism). More in detail, the Switch Transformer, a modified neural network architecture, has achieved notable performance efficiency in text understanding, which is a crucial task in natural language processing. Table 1 summarizes the recent existing approaches that indicate the originality of this extended work. The following section provides an overview of the Switch transformer model and its underlying principles, followed by a detailed description of our proposed architecture.

3. Methods and materials

This section introduces the design of our proposed Arabic Text Simplification model, ATSimST. Fig. 1 provides a comprehensive overview of the system's architecture, illustrating its four key phases (A-D). The ATSimST uses an Arabic punctuation detector for text segmentation (PDTS (Alshanqiti et al., 2022), as described in (B), and a modified T5 Transformer architecture (Raffel et al., 2020) for text-to-text rephrasing in (C). In attempting to segment a given input text into the shortest possible sequence of straightforward independent-clause sentences, the preprocessing tool (PDTS) is utilized. The purpose of the rephrasing phase (see (C), which presents the main contribution of this paper) is to produce a more understandable text by further simplifying the concatenated simple sentences. As an illustration, consider an input text denoted as X that comprises intricate sentences. Using rephrasing, ATSimST endeavours to decompose it into more straightforward sentences (S), such that S ← ATSimST(X) and $S = (s_1, s_2, \dots, s_{|n|})$, where n represents the number of simple sentences generated. As with most of the NLP applications, ATSimST begins with simple text preprocessing. This consists of removing diacritizations/noise and applying soft normalization. This preprocessing phase ensures that the input texts are cleansed without losing sentence structures, and more significantly, it partially retains the intended meaning. To implement this phase, we utilize CAMeL (CAMeL is an Arabic natural language processing tool: camel-tools.readthedocs.io), a publicly available state-of-the-art Arabic preprocessing Toolkit built upon Python. In the subsequent sections, we elaborate in greater depth on the core phases of ATSimST, laid out in (B) and (C) of Fig. 1.

3.1. Proposed switch transformer architecture

This paper proposes a Switch Transformer layer to improve the computational efficiency of Transformer models by using sparse activation. The main difference from the original Transformer is that instead of a single feed-forward neural network (FFN), the Switch layer includes multiple FFNs called "experts" (Fedus et al., 2022). Specifically, the Switch layer has two main components: (1) a set EX of expert networks, where $EX = \{ex_1, \cdots, ex_k\}$, $k \in \mathrm{N}$, and (2) a gating network GN that outputs a sparse n-dimensional vector. The EX are neural networks, each possessing its own set of parameters. This layer receives an input token x (from the previous Add-and-Norm layers) and then selects the best top-k experts (ex_i) as the next specific processing layers using Eq. 1. The gated probability g_i value for ex_i is calculated by Eq. 2, where w_r is a router parameter that is calculated based on the layers presented in the hidden states. The output of expert layers is determined by linearly weighing each calculation ex_i of the input token with its corresponding gate probability g_i, and then summing the results in out, see Eq. 3.

$$top - k \leftarrow \{E_i(x)\}_{i=1}^{N} \quad (1)$$
$$g_i(x) = \frac{e^{(w_r.x)i}}{\sum_j^N e^{(w_r.x)j}} \quad (2)$$
$$out = \sum_{i \in top-k} g_i(x).E_i(x) \quad (3)$$

In this paper, we examine the routing of input tokens to top-k experts as well as to a single expert with the highest level of confidence, determined by the maximum value of g_i.

Table 1: A summary of reviewed text simplification approaches that apply split-and-rephrasing techniques. Certain portions of this text are derived from our previous work (Alshanqiti et al., 2023)

Approach	Split technique	Rephrase technique	Transformer architecture
Context-preserving text simplification (Niklaus et al., 2019)	Defining a 35-handcrafted transformation rules	Semantic hierarchy of minimal propositions	✗
Fact-aware sentence split and rephrase with permutation invariant training (Guo et al., 2020)	Training a CNN model for predicting sentence split	Seq2seq Model	✗
Controllable text simplification with explicit paraphrasing (Maddela et al., 2020)	DisSim: A structural simplification tool consisting of 35 handcrafted grammar rules	Transformer seq2seq Model	✓
A memory-based sentence split and rephrase model (Fan et al., 2020)	Encoder and decoder transformer model connected by a memory gate layer		✓
Hierarchical generation for sentence simplification (Wang et al., 2019)	A semantic separator layer in the decoder model	LSTM Seq2seq-based Model	✗
BART model (Urakawa et al., 2024)	Text to Text Transformer model	Transformer seq2seq Model	✓
encoder-decoder transformer-based (SeqDiffuSeq) (Yuan et al., 2024)	Encoder-decoder transformer	Transformer seq2seq Model	✓
TSimAr (Alshanqiti et al., 2023)	PDTS (built upon mBERT model) (Alshanqiti et al., 2023)	FNet Text-to-Text based model	✓
Our ATSimST	PDTS (built upon mBERT model) (Alshanqiti et al., 2023)	Text-to-text model based on the switch transformer mechanism	✓

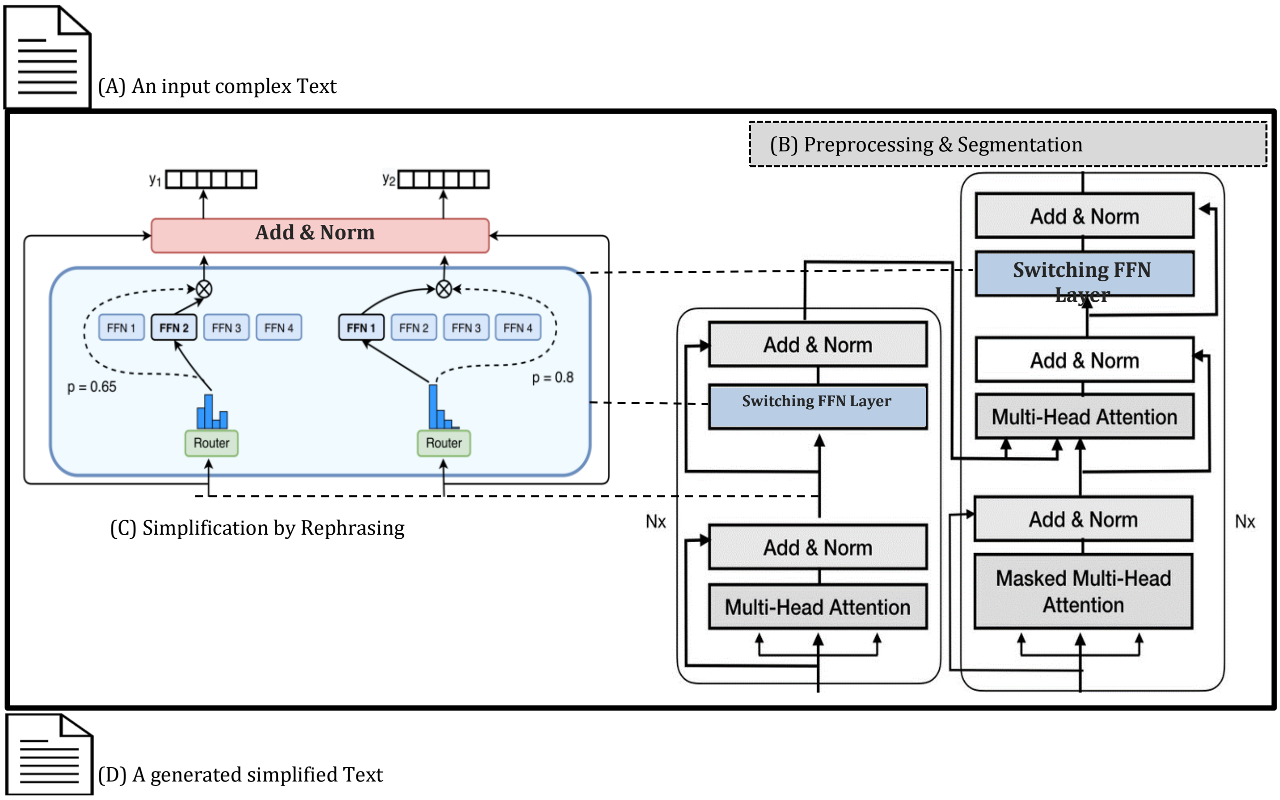

Fig. 1: A workflow illustration of the proposed Arabic text simplification tool (ATSimST) to streamline the process of simplifying Arabic textual documents

3.2. Text segmentation

The suggested ATSimST is built upon PDTS (Alshanqiti et al., 2022), which utilizes a pre-trained multilingual BERT model to detect missing punctuations. PDTS is an Arabic text-splitting tool that segments input texts into a series of possibly independent clauses. To elaborate, PDTS utilizes mBERT (github.com/google-research/bert/blob/master/multilingual.md) to make predictions on the appropriate punctuation marks that can be used as delimiters to split text. It queries mBERT a total of $|X|$ times to generate pu as follows in Eq. 4:

$$p(pu_i^m | t_i^m; pun, \theta) = PDTS \circ mBERT\ (X_i') \quad (4)$$
$$X_i' \leftarrow insertMaskToken(X, i), \forall i \in X \quad (5)$$

where, $pu^m{}_i$ denotes the accepted prediction by mBERT for a particular white-space at index i, $t^m{}_i$ represents the actual output of mBERT, pun and θ are user-defined model parameters used to filter out $t^m{}_i$, and X_i' is the input X with the ($MASK$) token inserted at index i. PDTS subsequently verifies the anticipated collection of punctuations $pu^m\ \forall i \in \{1,2, \cdots, |X|\}$ by employing four linguistic principles in a strategy resembling greediness (Alshanqiti et al., 2022). This paper focuses on the utilization of primary splitting punctuations, such as Comma, Colon, Semicolon, and Full-stop.

3.3. Text simplification by rephrasing

Inspired by the Transformers-based encoder-decoder architecture (Vaswani et al., 2017), which has demonstrated remarkable advancements in complex natural language processing tasks, we examine one of its optimized sequence-to-sequence models (i.e., text-to-text models). Specifically, we employ a streamlined variant of a T5-based model that replaces the standard single FFN layer with a switch layer (Fedus et al., 2022), indicated in Fig. 1. The first version of the T5-based model (Raffel et al., 2020) closely conforms to the original encoder-decoder Transformer architecture suggested in Vaswani et al. (2017). Initially, a series of tokens provided as input is transformed into a sequence of embeddings. These embeddings are subsequently fed into the encoder. The encoder comprises a stack of blocks, each including two subcomponents: a self-attention layer and a feed-forward network. The input of each subcomponent performs a streamlined kind of layer normalization, wherein the activations are solely rescaled without the inclusion of any additive bias. Following layer normalization, a residual skip connection (Raffel et al., 2020) combines the input of each subcomponent with its output. In addition, the dropout is implemented in several parts of the feedforward network, including the skip connection, attention weights, and at the input and output of the entire block. The decoder shares a similar structure to the encoder, but with adding a conventional attention mechanism after each self-attention layer. This attention mechanism focuses on the output of the encoder. The decoder's self-attention mechanism employs autoregressive or causal self-attention, restricting the model's attention to previous outputs. The output of the last decoder block is passed into a dense layer that has a softmax output. The weights of this layer are shared with the input embedding matrix. The Transformer architecture employs separate heads to handle attention mechanisms, with their outputs being combined through concatenation and subsequently processed.

In this paper, the goal of extending the standard T5 model with *N* expert FeedForward Neural Networks (FFNs) is to assess its proficiency in paraphrasing Arabic texts. Consequently, we examine downstream performance on an Arabic text-to-text benchmark used for text simplification, discussed in the next section.

4. Experimental implementation and results

In this section, we evaluate the efficacy of ATSimST and compare its rephrasing component with the main Arabic pre-trained text-to-text generation models. In our evaluation processes, we utilise ATSC (Alshanqiti et al., 2023), which is an Arabic corpus for text simplification. To measure the quality of the generated simplifications, we provide several automatic text matching metrics, such as SARI, which is the main metric for text simplification and rephrasing.

4.1. Dataset and experiment setup

Arabic Text Simplification Corpus (ATSC) We utilised ATSC (Alshanqiti et al., 2023), a benchmark corpus of Arabic text simplification samples, comprising 500 pairs of small-to-large complex source texts and their corresponding reference simplified texts, which human experts have segmented and rephrased. The ATSC has been constructed from different Arabic articles, encompassing a wide range of subjects, including history, geography, health, education, and technology. The statistical descriptions of ATSC are shown in Table 2.

Table 2: Statistical descriptions of ATSC corpus

	Original source texts	Simplified rewrites
# No. documents	500	500
# No. sentences	546	1243
# No. words (distinct)	17069 (6046)	17564 (5479)
# No. of vocabularies (for both complex and simplified texts) 6737		

1. Baselines: We compare the text rephrasing part in our ATSimST against (1) our previous model TSimAr (i.e., relying on FNet model), (2) the related pre-trained Arabic monolingual (Arabic-T5-small (Almarjeh, 2022), Arabic-T5, UBC-AraT5 (Nagoudi et al., 2022)), and (3) the available multilingual (MT5-base (Xue et al., 2020), mBART-large-50 (Tang et al., 2020)) models for text generation tasks. These text-to-text generating models are derived from the T5 encoder-decoder architecture transformer model (Raffel et al., 2020), except for mBART, which is a multilingual Sequence-to-Sequence model primarily used for text translation.
2. Automatic Metrics: To assess the effectiveness of the generated simplification y (denoted as Y ← ATSimST(st)), we measure its performance against a gold-reference gr (a simplified version created by human specialists). This evaluation is conducted using several automated measures.

- BLEU (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 (Papineni et al., 2002).
- ROUGE (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).
- METEOR (Metric for Evaluation for Translation with Explicit Ordering) is like 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 (Banerjee and Lavie, 2005).
- TER (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 (Snover et al., 2006).
- SARI (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 (Xu et al., 2016).

For BLEU, METEOR, ROUGE, and SARI, greater scores are indicative of superior quality that aligns with logical human evaluations. Conversely, a lower TER measure, which refers to lower edit-rating scores, suggests superior performance.

3. Implementation Details: To train and configure the text rephrasing component of our ATSimST, we applied a random split of (50%, 20%, 30%) on the ATSC corpus to generate train, dev, and test sets, respectively. In addition, we employed the Adam optimization method for training, utilizing a learning rate of 0.001. The training loop consists of 8k epochs, each with a batch size of 64 and a maximum sequence length of 256. Additionally, the text-to-text generation models, which are regarded as baselines in this paper, are publicly accessible at the Hugging Face (huggingface.co) under the model (card) names: 'google/mt5-base',

'facebook/mbart-large-50', 'flax-community/arabict5-small', 'UBC-NLP/AraT5-base-title-generation', and 'malmarjeh/t5-arabic-text-summarization'. In addition to employing NLTK (nltk.org) and CAMeL for text preprocessing, we have implemented the baseline models using the PyTorch (pytorch.org) framework. All experiments were conducted on a personal computer with an Intel i9 CPU, 64GB of RAM, and an NVIDIA GeForce RTX 3070 GPU.

4.2. Comparative analysis with related simplification approaches

Table 3 presents the automatic evaluation results of our proposed model, ATSimST, in comparison with several baseline models on the task of Arabic text simplification. Our proposed model, ATSimST, performs better across multiple evaluation metrics. Specifically, ATSimST achieves the highest SARI score of 0.81, outperforming the next best model, TSimAr, which has a score of 0.73. The SARI metric evaluates the quality of simplifications by comparing the model output against the original and reference sentences; a higher score indicates better simplification quality. This significant improvement in SARI suggests that ATSimST is more effective at generating simplified Arabic text while preserving the original meaning.

Regarding the BLEU score, which measures the overlap between the generated text and reference translations, ATSimST attains the highest score of 0.69, surpassing TSimAr's score of 0.65 and mBART's score of 0.40. This indicates that the outputs from ATSimST are more closely aligned with the reference simplifications, reflecting higher accuracy and fluency in the simplified text generated by our model. For the METEOR metric, which accounts for synonymy and morphological variations, ATSimST again achieves the highest score of 0.72, compared to TSimAr's 0.68 and mBART's 0.61. The superior METEOR score emphasizes ATSimST's ability to produce semantically appropriate simplifications closer to human references.

Concerning the ROUGE metrics, ATSimST obtains the highest scores in ROUGE2 and ROUGE-L, with values of 0.72 and 0.76, respectively. These metrics evaluate the overlap of bigrams (ROUGE-2) and the longest common subsequence (ROUGE-L) between the generated text and reference, indicating that ATSimST more effectively captures the essential content of the original text in its simplifications. While mBART has a higher ROUGE-1 score of 0.72 (indicated by an asterisk), ATSimST achieves a higher ROUGE-1 score of 0.78, suggesting a potential error in asterisk placement. Our model has a higher ROUGE-1 score, reinforcing its capability to retain key unigrams from reference texts. In contrast, TSimAr achieves the lowest TER, which measures the amount of editing needed to match the reference text (lower scores indicating better performance), at 0.34. Although ATSimST's TER score is 0.93, which is higher than TSimAr's, it remains competitive with other models like Arabic-T5-small (0.89) and indicates acceptable performance in this metric.

Execution time is another crucial factor, particularly for practical applications. The fastest model is Arabic-T5-small, with an execution time of 128 seconds. ATSimST has an execution time of 421 seconds, which is longer than some baseline models but considerably shorter than mBART's 3685 and TSimAr's 724 seconds. While execution time is an important consideration, the substantial and undeniable improvements in simplification quality offered by ATSimST reassure us of its performance, which may justify the additional computational resources required. Moreover, the Switch Transformer layer offers a small but worthwhile improvement in execution time (approximately 303 seconds) compared to the FNet layer used in TSimAr (Alshanqiti et al., 2023).

To sum up, the experimental results illustrate that ATSimST generally outperforms existing models in generating simplified Arabic text. The improvements in SARI, BLEU, METEOR, and ROUGE metrics highlight the model's effectiveness in producing high-quality, semantically accurate simplifications that closely align with human-generated references. These findings not only validate the effectiveness of ATSimST but also underscore its potential to revolutionize the field of Arabic text simplification.

Table 3: Automatic evaluation results. Certain portions of these results are derived from our previous work (Alshanqiti et al., 2023)

					ROUGE			
	SARI	BLEU	TER	METEOR	R-1	R-2	R-L	ET (Sec)
Arabic-T5-small (Almarjeh, 2022)	0.23	0.02	0.89	0.11	0.23	0.11	0.22	128*
Arabic-T5 (huggingface.co, 2022)	0.21	0.01	0.94	0.05	0.14	0.04	0.14	145
MT5-base (Xue et al., 2020)	0.18	0.00	0.98*	0.02	0.05	0.01	0.04	213
mBART (Tang et al., 2020)	0.53	0.40	0.52	0.61	0.72*	0.53	0.70	3685
UBC-AraT5 (Nagoudi et al., 2022)	0.21	0.00	0.95	0.05	0.12	0.04	0.11	178
TSimAr (Alshanqiti et al., 2023)	0.73	0.65	0.34	0.68	0.71	0.67	0.71	724
Our ATSimST	0.81*	0.69*	0.93	0.72*	0.78	0.72*	0.76*	421

The best performance for each metric is indicated by an asterisk (*)

5. Conclusion

This paper presented ATSimST, a split-and-rephrase approach to Arabic text simplification based on a Switch Transformer architecture coupled with a preprocessing segmentation step. ATSimST integrates a punctuation-based segmentation module (PDTS) built on mBERT to split complex text into more straightforward sentences. These sentences are then rephrased using an efficient

Switch Transformer model. Experimental results on the ATSC dataset demonstrate that ATSimST outperforms existing state-of-the-art models, achieving substantial improvements in SARI, BLEU, METEOR, ROUGE scores, and, more importantly, in execution time compared to TSimAr. These results highlight the effectiveness of ATSimST in generating high-quality and semantically accurate simplified Arabic text. While ATSimST still exhibits a longer execution time than some baseline models, its acceptable simplification quality justifies the additional computational cost.

Despite the continued improvements in Arabic text simplification approaches, they still face several challenges, such as the lack of large-scale parallel corpora, the diversity of Arabic dialects, and the language's complex linguistic structure.

This work advances Arabic text simplification research and opens several promising directions for future investigation. Future work will explore incorporating additional linguistic features into the model, optimizing the preprocessing segmentation process, and investigating the impact of different Transformer architectures on model performance. Furthermore, we plan to investigate the applicability of ATSimST to different Arabic dialects, aiming to broaden its impact and contribute to greater accessibility in Arabic NLP applications.

List of abbreviations

NLP	Natural language processing
NLU	Natural language understanding
ATS	Automated text simplification
LS	Lexical simplification
TG	Text generation
FFN	Feed-forward network
ATSimST	Arabic text simplification using switch transformer
TSimAr	Transformer-based simplification for Arabic
PDTS	Punctuation detector for text segmentation
mBERT	Multilingual BERT
CAMeL	Computational analysis of mediated language
T5	Text-to-text transfer transformer
BART	Bidirectional and auto-regressive transformers
BLEU	Bilingual evaluation understudy
ROUGE	Recall-oriented understudy for gisting evaluation
METEOR	Metric for evaluation of translation with explicit ordering
TER	Translation edit rate
SARI	System output against references and against the input sentence
GN	Gating network
EX	Expert networks
MT5	Multilingual T5
mBART	Multilingual BART
LSTM	Long short-term memory
NLTK	Natural language toolkit
ATSC	Arabic text simplification corpus
Seq2seq	Sequence to sequence
R-1	ROUGE-1 (unigram overlap)
R-2	ROUGE-2 (bigram overlap)
R-L	ROUGE-L (longest common subsequence overlap)
ET	Execution time
k	An integer parameter

Compliance with ethical standards

Conflict of interest

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

References

Alkaldi W and Inkpen D (2023). Text simplification to specific readability levels. Mathematics, 11(9): 2063. https://doi.org/10.3390/math11092063

Almarjeh MB (2022). An Arabic abstractive text summarization model. Available online at: https://huggingface.co/malmarjeh/mbert2mbert-arabic-text-summarization

Alshanqiti A, Alkhodre A, Namoun A, Albouq S, and Nabil E (2023). A transformer seq2seq model with fast Fourier transform layers for rephrasing and simplifying complex Arabic text. International Journal of Advanced Computer Science and Applications, 14(2): 888-898. https://doi.org/10.14569/IJACSA.2023.01402101

Alshanqiti A, Namoun A, Alsughayyir A, Mashraqi AM, Gilal AR, and Albouq SS (2021). Leveraging DistilBERT for summarizing Arabic text: an extractive dual-stage approach. IEEE Access, 9: 135594-135607. https://doi.org/10.1109/ACCESS.2021.3113256

Alshanqiti AM, Albouq S, Alkhodre AB, Namoun A, and Nabil E (2022). Employing a multilingual transformer model for segmenting unpunctuated Arabic text. Applied Sciences, 12(20): 10559. https://doi.org/10.3390/app122010559

Al-Thanyyan SS and Azmi AM (2023). Simplification of Arabic text: A hybrid approach integrating machine translation and transformer-based lexical model. Journal of King Saud University-Computer and Information Sciences, 35(8): 101662. https://doi.org/10.1016/j.jksuci.2023.101662

Alva-Manchego F, Scarton C, and Specia L (2020). Data-driven sentence simplification: Survey and benchmark. Computational Linguistics, 46(1): 135-187. https://doi.org/10.1162/coli_a_00370

Banerjee S and Lavie A (2005). METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In the Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization, Association for Computational Linguistics, Ann Arbor, Michigan, USA: 65-72.

Espinosa-Zaragoza I, Abreu-Salas J, Lloret E, Pozo PM, and Palomar M (2023). A review of research-based automatic text simplification tools. In the Proceedings of the 14th International Conference on Recent Advances in Natural Language Processing, INCOMA, Varna, Bulgaria: 321-330. https://doi.org/10.26615/978-954-452-092-2_036

Fan X, Liu Y, Liu G, and Su B (2020). A memory-based sentence split and rephrase model with multi-task training. In the Neural Information Processing: 27th International Conference, Springer International Publishing, Bangkok, Thailand: 643-654. https://doi.org/10.1007/978-3-030-63830-6_54

Fedus W, Zoph B, and Shazeer N (2022). Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. Journal of Machine Learning Research, 23(120): 1-39.

Gamal D, Alfonse M, Jiménez-Zafra SM, and Aref M (2022). Survey of Arabic machine translation, methodologies, progress, and challenges. In the 2nd International Mobile, Intelligent, and

Ubiquitous Computing Conference, IEEE, Cairo, Egypt: 378-383. https://doi.org/10.1109/MIUCC55081.2022.9781776

Gooding S and Kochmar E (2019). Recursive context-aware lexical simplification. In the Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, Association for Computational Linguistics, Hong Kong, China: 4853-4863. https://doi.org/10.18653/v1/D19-1491

Guo Y, Ge T, and Wei F (2020). Fact-aware sentence split and rephrase with permutation invariant training. In the Proceedings of the AAAI Conference on Artificial Intelligence, 34(5): 7855-7862. https://doi.org/10.1609/aaai.v34i05.6291

Hao T, Li X, He Y, Wang FL, and Qu Y (2022). Recent progress in leveraging deep learning methods for question answering. Neural Computing and Applications, 34: 2765–2783. https://doi.org/10.1007/s00521-021-06748-3

Horn C, Manduca C, and Kauchak D (2014). Learning a lexical simplifier using Wikipedia. In the Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Baltimore, USA, 2: 458-463. https://doi.org/10.3115/v1/P14-2075

Karim AA, Usama M, Ibrahim MA, Hatem Y, Wael W, Mazrua AT, El-Monayer GK, Elbanhawy M, Foad K, Moawad IF (2024). Arabic abstractive summarization using the multilingual T5 model. In the 6th International Conference on Computing and Informatics, IEEE, Cairo, Egypt: 223-228. https://doi.org/10.1109/ICCI61671.2024.10485135

Maddela M and Xu W (2018). A word-complexity lexicon and a neural readability ranking model for lexical simplification. In the Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Brussels, Belgium: 3749–3760. https://doi.org/10.18653/v1/D18-1410

Maddela M, Alva-Manchego F, and Xu W (2020). Controllable text simplification with explicit paraphrasing. Arxiv Preprint Arxiv:2010.11004. https://doi.org/10.48550/arXiv.2010.11004

Martin L, Fan A, De La Clergerie E, Bordes A, and Sagot B (2020). MUSS: Multilingual unsupervised sentence simplification by mining paraphrases. Arxiv Preprint Arxiv:2005.00352. https://doi.org/10.48550/arXiv.2005.00352

Nagoudi EMB, Elmadany A, and Abdul-Mageed M (2022). AraT5: Text-to-text transformers for Arabic language generation. In the Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Dublin, Ireland, 1: 628–647. https://doi.org/10.18653/v1/2022.acl-long.47

Niklaus C, Cetto M, Freitas A, and Handschuh S (2019). Transforming complex sentences into a semantic hierarchy. Arxiv Preprint Arxiv:1906.01038. https://doi.org/10.48550/arXiv.1906.01038

Niklaus C, Cetto M, Freitas A, and Handschuh S (2021). Context-preserving text simplification. Arxiv Preprint Arxiv:2105.11178. https://doi.org/10.48550/arXiv.2105.11178

North K, Ranasinghe T, Shardlow M, and Zampieri M (2024). MultiLS: A multi-task lexical simplification framework. Arxiv Preprint Arxiv:2402.14972. https://doi.org/10.18653/v1/2024.tsar-1.1 **PMCid:PMC10796270**

Papineni K, Roukos S, Ward T, and Zhu WJ (2002). BLEU: A method for automatic evaluation of machine translation. In the Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, Philadelphia, USA, 311-318. https://doi.org/10.3115/1073083.1073135

Qiang J and Wu X (2019). Unsupervised statistical text simplification. IEEE Transactions on Knowledge and Data Engineering, 33(4): 1802-1806. https://doi.org/10.1109/TKDE.2019.2947679

Raffel C, Shazeer N, Roberts A, Lee K, Narang S, Matena M, Zhou Y, Li W, Liu PJ (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. Journal of Machine Learning Research, 21(140): 1-67.

Snover M, Dorr B, Schwartz R, Micciulla L, and Makhoul J (2006). A study of translation edit rate with targeted human annotation. In the Proceedings of the 7th Conference of the Association for Machine Translation in the Americas: Technical Papers, Association for Machine Translation in the Americas, Cambridge, USA, 223-231.

Stajner S and Saggion H (2018). Data-driven text simplification. In the Proceedings of the 27th International Conference on Computational Linguistics: Tutorial Abstracts, Association for Computational Linguistics, Santa Fe, USA: 19–23. https://doi.org/10.1093/oxfordhb/9780199573691.013.52 **PMCid:PMC5985336**

Surya S, Mishra A, Laha A, Jain P, and Sankaranarayanan K (2019). Unsupervised neural text simplification. In the Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Florence, Italy: 2058-2068. https://doi.org/10.18653/v1/P19-1198

Tang Y, Tran C, Li X, Chen PJ, Goyal N, Chaudhary V, Gu J, Fan A (2020). Multilingual translation with extensible multilingual pretraining and finetuning. Arxiv Preprint Arxiv:2008.00401. https://doi.org/10.48550/arXiv.2008.00401

Urakawa T, Taguchi Y, Niitsuma T, and Tamori H (2024). A Japanese news simplification corpus with faithfulness. In the Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation, ELRA and ICCL, Torino, Italia: 659-665.

Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Ł, Polosukhin I (2017). Attention is all you need. Advances in Neural Information Processing Systems, 30: 5998–6008.

Wang M, Ozaki H, Koreeda Y, and Yanai K (2019). Split first and then rephrase: Hierarchical generation for sentence simplification. In the International Conference of the Pacific Association for Computational Linguistics, Springer Singapore, Hanoi, Vietnam: 15–27. https://doi.org/10.1007/978-981-15-6168-9_2

Xu W, Napoles C, Pavlick E, Chen Q, and Callison-Burch C (2016). Optimizing statistical machine translation for text simplification. Transactions of the Association for Computational Linguistics, 4: 401-415. https://doi.org/10.1162/tacl_a_00107

Xue L, Constant N, Roberts A, Kale M, Al-Rfou R, Siddhant A, Barua A, Raffel C (2020). mT5: A massively multilingual pre-trained text-to-text transformer. Arxiv Preprint Arxiv:2010.11934. https://doi.org/10.18653/v1/2021.naacl-main.41 **PMid:33576803 PMCid:PMC8154059**

Yuan H, Yuan Z, Tan C, Huang F, and Huang S (2024). Text diffusion model with encoder-decoder transformers for sequence-to-sequence generation. In the Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 1: 22-39. https://doi.org/10.18653/v1/2024.naacl-long.2

Zhou M, Duan N, Liu S, and Shum HY (2020). Progress in neural NLP: Modeling, learning, and reasoning. Engineering, 6(3): 275-290. https://doi.org/10.1016/j.eng.2019.12.014