Communications in Mathematics and Applications
Vol. 17, No. 2, pp. 289–311, 2026
ISSN 0975-8607 (online); 0976-5905 (print)
Published by RGN Publications
DOI: 10.26713/cma.v17i2.3662

http://www.rgnpublications.com

Research Article

Domain Adaptation for Bilingual Arabic-English Neural Machine Translation (NMT)

Dalal Abduallah Aljohany*[1], Hassanin M. Al-Barhamtoshy[2], Felwa Abukhodair[1] and Maysoon Abulkhair[1]

[1] *Faculty of Computing and Information Technology, King Abdulaziz University Jeddah, Abdullah Alsulaiman St., Al-Jamiah, Jeddah 21589, Saudi Arabia*
[2] *International Jeddah Collage, Ibn Rasheed Al Fehri, Taiba, Jeddah 23831, Saudi Arabia*
***Corresponding author:** daljohany0001@stu.kau.edu.sa

Received: March 15, 2026 **Revised:** May 2, 2026 **Accepted:** May 10, 2026

Abstract. The importance of *Arabic Machine Translation* (AMT) cannot be overstated as it seeks to enhance communication between Arabic and non-Arabic speakers. Given that there are an estimated 300 million speakers globally, Arabic is essential for cultural exchanges, global commerce, and diplomatic relations. Nevertheless, what creates substantial obstacles for the Arabic language with regards to *Machine Translation* (MT) is the fact that it is categorized as a language with low resources as well as one with complex and rich morphological features. *Neural Machine Translation* (NMT) has emerged as the pinnacle of MT methodologies. Although there exist numerous NMT tools and models designed for translation of Arabic text, the translation quality is often lacking, particularly with texts that fall out-of-domain. A primary technical obstacle associated with AMT stems from the scarcity of bilingual datasets for such texts that are out-of-domain. Consequently, this research seeks to enhance AMT across various domains. To achieve this, we designed a multi-domain parallel corpus specifically for *Arabic-English* (AR-EN) translation, which was later utilized to train the proposed NMT model. The model uses the encoder–decoder architecture that employs *Long Short-Term Memory* (LSTM) networks, which are themselves integrated with attention mechanisms. The experimental results show that this proposed model improves accuracy of translation and reduces the loss. To evaluate the efficacy of our proposed NMT model we compared its performance against both OPUS MT system and Google Translate across four domains using BLEU as a metric for assessment. In translations from Arabic to English, our model consistently outperformed both OPUS MT and Google Translate across every domain assessed—averaging +18.66 BLEU points above OPUS MT while exceeding Google Translate by approximately +18.11 BLEU points overall. Similarly consistent superiority was observed in the English to Arabic direction where average gains reached +22.39 points over OPUS MT and +15.36 points over Google Translate.

Keywords. Arabic Machine Translation (AMT), Natural Language Processing (NLP), Machine Translation (MT), Neural Machine Translation (NMT), Parallel corpus

Mathematics Subject Classification (2020). 68T50, 68T05, 68U15

Copyright © 2026 Dalal Abduallah Aljohan, Hassanin M. Al-Barhamtoshy, Felwa Abukhodair and Maysoon Abulkhair. *This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.*

1. Introduction

Machine Translation (MT) is a specialized area within the field of *Natural Language Processing* (NLP) that is essential for the advancement of computer-based translation systems (Qun and Xiaojun [25]). The primary goal of MT is to automate translation between texts in various languages, while preserving the source text's meaning and stylistic nuances (El Idrysy *et al*. [14]). However, designing a precise MT system that delivers translations of high quality remains a formidable task due to the intricate and structurally rich nature of natural languages.

Over the years, researchers have introduced various MT methods. At first, translators utilized linguistically based techniques, which led to the development of *Rule-Based Machine Translation* (RBMT). However, the limitations of RBMT systems arise from their dependence on rules that are constructed manually. As such, the development of new translation modules for different languages has become even more complex (Costa-Jussà *et al*. [9], and Thurmair [29]). In response to the scalability issues of RBMT, researchers developed *Statistical Machine Translation* (SMT) as well as *Example-Based Machine Translation* (EBMT) (Moussallem *et al*. [22]). This is due to the fact that scalability is a crucial requirement in MT systems, since they need to process data at a web-scale. While these methods tackled the limitations of RBMT, challenges that had been previously addressed through linguistics-based techniques reemerged; the majority of which were associated with ambiguity. In response, researchers integrated RBMT and SMT through a *Hybrid Machine Translation* (HMT) approach to address the limitations of each individual approach. Although hybrid approaches typically yield impressive outcomes, they still retain some of the issues inherent in RBMT (e.g., the considerable effort required to implement new rules to address specific syntax divergences). A more recent MT approach was introduced. This approach, *Neural Machine Translation* (NMT), relies on *Neural Network* (NN) algorithms and has produced outstanding results and has become a widely adopted MT method (Moussallem *et al*. [22]).

Over the past few years, the fast-paced development of deep learning methods and their integration in MT has resulted in significant progress across various translation tasks. NMT systems, which have been trained on extensive corpora, have exhibited remarkable efficacy in translating generic language (Alghamdi *et al*. [2]). Nonetheless, such models frequently underperform on data that is out-of-domain data (Koehn and Knowles [20]), particularly if the target domain features a unique vocabulary and style (Daumé III and Jagarlamudi [11]).

The Arabic language ranks among the most extensively spoken languages worldwide, boasting more than 300 million native speakers and being designated one of the official languages of the United Nations (Aransa [5]). Though there are numerous MT approaches and tools available for Arabic text translation, the translation quality is often inadequate. Within the context of *Arabic Machine Translation* (AMT), researchers encounter specific issues that require attention. As a Semitic language, Arabic is characterized by its rich and intricate morphological structure, which distinguishes it from other languages (e.g., French and English) (Shalhoub-Awwad and Leikin [26]). It is this complexity in morphological structure, along with the Arabic language's other linguistic features, that complicates the process of automated translation to and from it.

Apart from the concerns noted above, there exists a deficiency of high-quality parallel corpora that is large enough to train or calibrate AMT across various domains

(Alghamdi *et al*. [2]). It is widely recognized that NMT systems struggle with translations that are domain-specific, particularly in low-resource languages, i.e., those with limited linguistic data (Koehn and Knowles [20]). To tackle these issues, researchers have explored domain adaption techniques to design AMT systems tailored to specific domains. Currently, AMT researchers mainly concentrate on translating a limited range of domains (e.g., official documents and news), with few efforts directed towards other domain-specific translations, e.g., financial (Alghamdi *et al*. [2]) and medical (Ehab [12]).

Many parallel data accessible to researchers has been confined to texts generated by parliamentary discussions and international organizations (Noor-e-Hira *et al*. [23]). Regrettably, current single-domain AMT techniques are not effective across multiple domains. As such, there is heightened need for multiple domain NMT techniques to address this shortcoming (Alghamdi *et al*. [2]). Hence, this study seeks to bridge the gap in enhancing bilingual Arabic-English machine translation in diverse domains. This study makes the following key contributions:

- It proposes an innovative NMT model that utilizes the encoder-decoder architecture based on LSTM networks integrated with attention mechanisms to accurately capture contextual dependencies within the input sequence.
- It constructs and preprocesses a multiple domain Arabic-English parallel corpus encompassing a variety of domains.
- It analyzes the performance of the proposed model across various domains employing standard evaluation metrics (e.g., BLEU score).
- It conducts extensive experiments focusing on both Arabic to English and English to Arabic translation endeavors spanning multiple domains; comparisons with robust models reveal substantial BLEU score improvements achieved by our proposed model—with gains exceeding +18.66 BLEU points against OPUS(ar-en) alongside 18.11 versus Google Translate within Arabic to English translations; similarly when translating English to Arabic, it achieves average BLEU score improvements of +22.39 points over OPUS (en-ar) and +15.36 points over Google Translate.

The remainder of this paper includes background information presented in Section 2; prior research relevant specifically towards AMT detailed within Section 3. The problem statement presented in Section 4. The discussion surrounding our proposed model architecture outlined via Section 5. The experimental setup overviewed briefly under Section 6 followed experimental results discussed throughout Section 7. The concluding remarks provided ultimately found within Section 8.

2. Background

2.1 Neural Machine Translation (NMT)

Neural Machine Translation (NMT) represents the latest advancement in MT methodologies, having emerged around the year 2018 (Gamal *et al*. [16]). NMT autonomously acquires the capability to translate text through the application of deep learning techniques, which facilitate more rapid translations compared to traditional SMT approaches and yield superior quality outcomes. The employment of vector representations, commonly referred to as embeddings, for both lexical items and internal states constitute its primary distinguishing

feature. NMT exemplifies the principles of *Sequence-to-Sequence* (Seq2Seq) learning (Yadav *et al*. [30]). The overarching architecture of Seq2Seq encompasses an encoder and a decoder (Sutskever *et al*. [28]). The encoder is responsible for generating a representation of the input source sequence of words into an intermediate sequence output, whereas the decoder utilizes the intermediate sequence outputs to construct the target language sequence of words. Various models are implemented, including *Recurrent Neural Networks* (RNN), *Convolutional Neural Networks* (CNN), *Long Short-Term Memory* (LSTM) networks, and their combinations (Singh *et al*. [27]).

NMT outperforms SMT, even though it needs massive amounts of parallel data for training parallel data for training (Zakraoui *et al*. [31]). This technique encodes a given source language text into a continuous vector (context vector) using RNN. It then uses the second RNN as the decoder for predicting words in the TL. Figure 1 illustrates the overall architecture of the most popular NMT system.

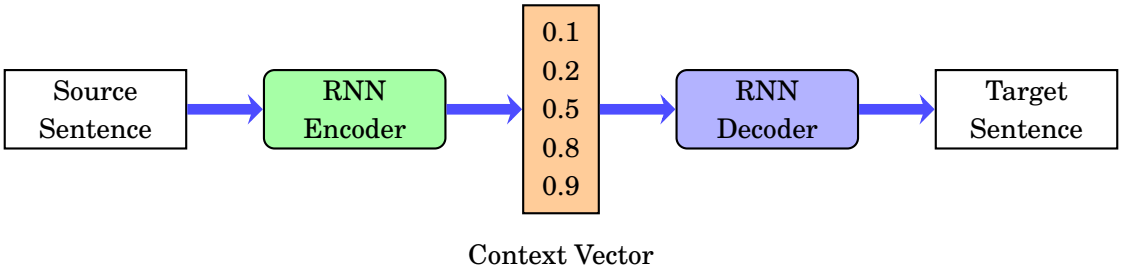

Figure 1. Overall architecture of the most popular NMT system

2.2 Attention Mechanism

Early NMT models depend on a fixed context vector — a single compressed representation of the entire source sentence produced by the encoder RNN. This vector was then used to initialize the decoder, which generated the full translation. However, this design had a clear limitation: it forced all the information from the source sentence into one fixed-length representation, making it difficult to handle longer or more complex sentences (Ataman *et al*. [6]). A major improvement came with the introduction of the attention mechanism. Instead of using a single static context vector, attention allows the model to compute a new context vector at each decoding step. This enables the decoder to focus on the most relevant parts of the source sentence when generating each target word (as illustrated in Figure 2). As a result, the translation process becomes more flexible and better aligned with the input, leading to improved performance (Bahdanau *et al*. [7], and Luong *et al*. [21]).

3. Related Works

The development of AMT has progressed through several major paradigms over time. Early approaches were primarily based on *Rule-Based Machine Translation* (RBMT) systems, which relied on handcrafted linguistic rules and dictionaries to generate translations. These systems were later followed by *Statistical Machine Translation* (SMT) methods, which leveraged probabilistic models and parallel corpora to learn translation patterns from data. More recently, the field has shifted toward *Neural Machine Translation* (NMT), which employs

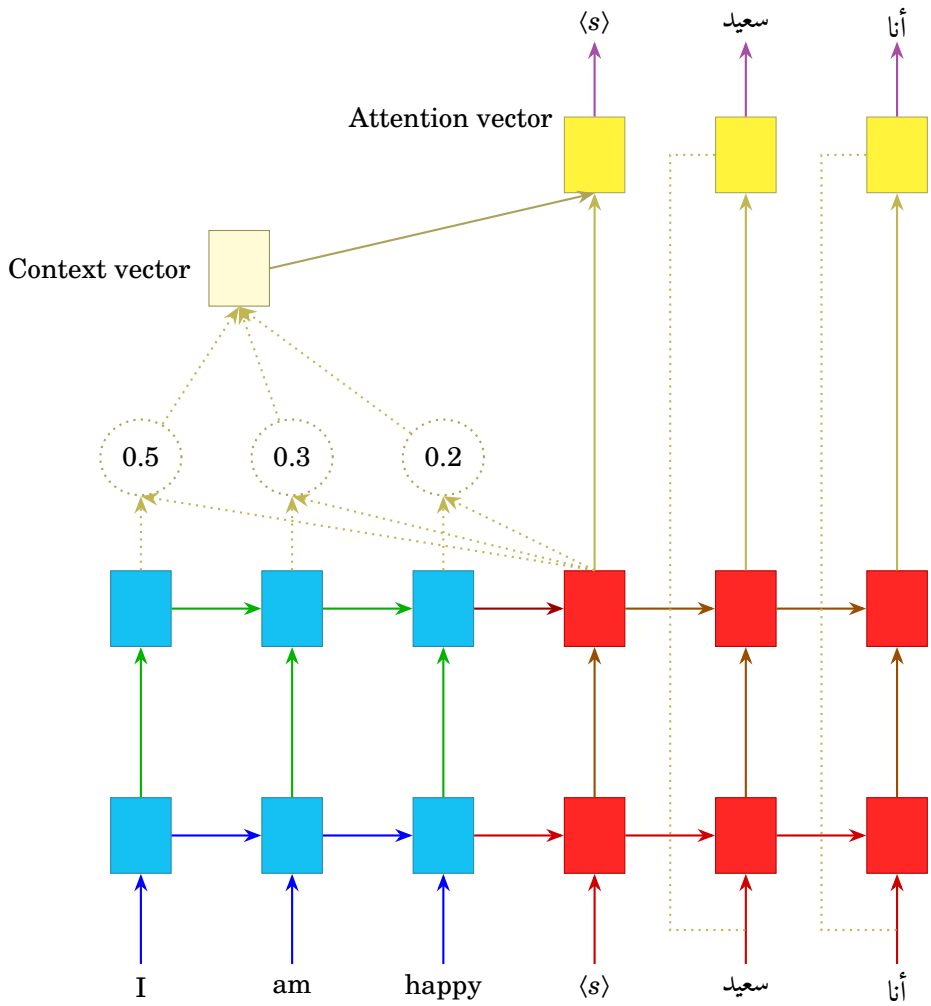

Figure 2. Encoder-Decoder with attention mechanism structure (El Idrysy *et al*. [14])

deep learning architectures to model the translation process in an end-to-end manner and has achieved significant improvements in translation quality.

One proposed system explained by Al-Taani and Hailat [4] focuses on direct English-Arabic RBMT system utilizing a dictionary. It categorizes the domain into two segments: a collection of up to 184 English proverbs sourced from Al-Mawrid, an English-Arabic dictionary, and a selection of 125 well-structured English sentences drawn from various textbooks. The researchers found their proposed system to provide an accuracy of 57.3% correct translations.

Abdullah [1] developed RBMT system aimed at translating lengthy noun phrases. In this case, Arabic serves as the *Source Language* (SL), while English is the *Target Language* (TL). They implemented a straightforward method to break down long sentences into smaller word segments, relying on various particles such as conjunctions, prepositions, and quantifiers for division. Each phrase in the source language was then translated separately before reconstructing the target language sentences by linking the translated phrases. The system was evaluated using 100 thesis titles comprising lengthy sentences from specific fields, namely management and economics, with assessments carried out by expert human translators. The results indicated high efficiency for most tested sentences.

Additionally, Farhat and Al-Taani [15] presented a RBMT system tailored for converting simple English declarative sentences into properly structured Arabic equivalents. The evaluation utilized a dataset consisting of 70 unique simple English declarative sentences created by language experts. The performance of this proposed system was compared against two commercial alternatives: Google Translator and Systran Systems. The proposed system recorded an accuracy rate of 85.71%, significantly outperforming Google's 31.42% and Systran's 20% on the same test set.

Ehab *et al*. [13] introduced an EBMT system designed to translate medical texts from English into Arabic. This system utilized 259 medical sentences sourced from internal medicine publications. The experimental findings reveal that the proposed system achieved a BLEU score of 48.6%.

In another study, Berrichi and Mazroui [8] introduced a SMT system aimed at translating from English to Arabic while developing an advanced alignment technique based on morphological pre-processing coupled with a bilingual dictionary as an auxiliary resource for enhancing certain alignments. Experimental outcomes demonstrated that integrating the bilingual dictionary significantly improved alignment quality, ultimately enhancing BLEU scores by 5%.

To further elevate English-to-Arabic MT quality, Khemakhem *et al*. [19] proposed an innovative approach focused on extracting and integrating classes about morpho-syntactic and semantic aspects within SMT frameworks. Combining semantic classes facilitated better translations where words held multiple meanings thereby enhancing overall translation quality — a strategic integration targeting lexical meaning disambiguation within lexical semantics yielded positive research outcomes reflected as a 1.3% increase in BLEU scores.

Almahairi *et al*. [3] performed a comparative study of NMT versus a traditional SMT system, concentrating on Arabic translations on both direction Arabic to English and English to Arabic. Their analysis involved extensive comparisons with various preprocessing configurations for Arabic script, revealing that both SMT and NMT achieve similar performance levels. They also found that effective preprocessing of the Arabic script positively influences both systems. However, NMT significantly surpasses SMT in the out-of-domain evaluation set (MEDAR), achieving a BLEU score of 24.46% in English-to-Arabic translation, while their performances were comparable in the in-domain evaluation set (MT05).

In another study, Gashaw and Shashirekha [17] examined the NMT for the Amharic to Arabic language pair due to limited data availability. They developed two models based on LSTM and GRU architecture, utilizing an attention-based encoder-decoder framework adapted from OpenNMT. For their experiments, they created a small parallel corpus from Quranic texts by modifying existing monolingual Arabic texts alongside corresponding Amharic translations available on Tanzile. The comparison between LSTM-based OpenNMT, GRU-based OpenNMT, and Google Translation revealed that LSTM outperformed both GRU and Google translation systems, achieving BLEU scores of 12%, 11%, and 6%, respectively.

Hamed *et al*. [18] employed NMT techniques to enhance the accuracy of translations from Arabic Holy Quran texts into Italian. They introduced two sophisticated deep learning architectures — LSTM and GRU — integrating an attention mechanism to facilitate this translation task. The effectiveness of these models was evaluated using BLEU and ROUGE metrics; experimental results showed that LSTM achieved average scores of 0.18 on BLEU and 0.17 on ROUGE while GRU reached averages of 0.17 on BLEU and 0.16 on ROUGE.

4. Problem Statement

Although there has been substantial advancement in the field of NMT, automatic MT continues to encounter considerable constraints associated with domain coverage as well as the availability of data. Most current studies have predominantly concentrated on a narrow range of domains. Consequently, those that are domain-specific, especially ones in specialized fields (e.g., IT, computer sciences, legal, and the sciences), are insufficiently explored and lack sufficient support. This disparity in domain representation results in a vital issue: models developed using data from a single domain frequently struggle to generalize well across various domains. As a result, current AMT systems demonstrate diminished performance when utilized with data from various domains or domain-shifted data. Thus, there is increasing demand for resilient multi-domain NMT techniques that can effectively manage heterogeneous data sources and enhance the quality of translation quality in a multitude of domains. As a result, there is great need for a comprehensive Arabic-English parallel corpus that prioritizes both source and target text quality while encompassing a wide array of text types and domains.

Most studies in the existing literature have focused on translation from English to Arabic. In contrast, the relatively minor focus on bilingual as well as Arabic to English studies reveals a significant gap. This underrepresentation could hinder global access to Arabic culture and literature. It also indicates a deficiency in research that concurrently investigates both languages or prioritizes bidirectional translation tools and corpora. Bilingual studies play a vital role in the progression of linguistic fields (e.g., NLP and MT), and in fostering bilingual education and cross-cultural communication. Without comprehensive research in this field, the development of effective solutions for smooth interaction between the Arabic and English languages is notably restricted.

For this reason, in this paper we developed a multiple domain parallel corpus for bilingual Arabic-English translation. We then used this corpus to train our proposed bilingual Arabic-English NMT that utilizes the encoder–decoder architecture based on LSTM networks integrated with attention mechanisms.

5. Proposed System Architecture

This paper employs an NMT methodology to design an automated bilingual translation system for English and Arabic. The comprehensive design is based on an encoder–decoder architecture, augmented with an attention mechanism to better grasp the contextual relationships present in each sequence of words. The encoder's role is to transform the source language's input sentences into continuous vector representations encoding semantic information. The decoder's role is generation of sentences into the target language by utilizing the encoded representations in conjunction with the attention mechanism that establishes token-to-token alignment. Overall, as shown in Figure 3, the system pipeline is composed of four key parts: data collecting, preprocessing, NMT and evaluating. In the first stage, parallel data is collected, and it consists of Arabic sentences along with their respective translations in English. Then, this data undergoes preprocessing to guarantee quality and preparedness for further processing. The resulting data is then converted into numerical formats suitable for input into the NMT model. The model's core is the LSTM-based NMT architecture that employs an

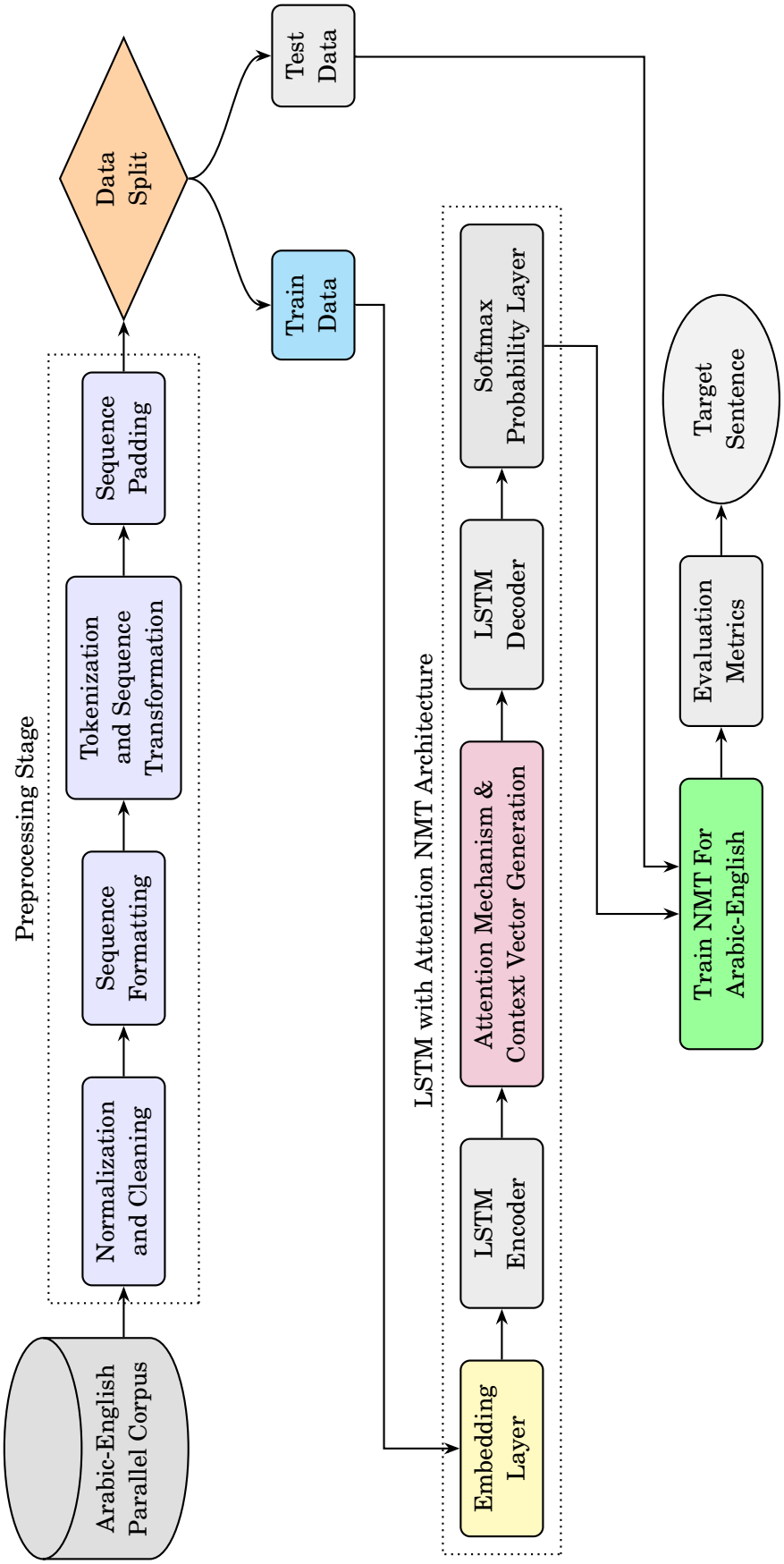

Figure 3. Architecture of proposed system

attention mechanism. Initially, input tokens are directed through an embedding layer designed to capture data about semantics and position. Following this, an LSTM encoder processes the embeddings, resulting in contextual representations of source sentences. The attention mechanism then calculates alignment scores and creates a context vector. This is what will enable the decoder to concentrate on significant sections while translating the input sequence. This vector is utilized by the LSTM decoder to produce the target sentence incrementally, while a Softmax layer generates the last probability distribution over the target vocabulary. The final stage consists of evaluating this trained NMT model against standard metrics on the test dataset to measure the quality of the translations.

5.1 Data Collection and Preparation

To compile the data, we accessed the deanship website of King Abdulaziz University (KAU) to obtain a total of 4,028 dissertations titles, both master's and doctoral. The titles were available in two languages Arabic and English version in University's website. The dissertation titles covered various domains (i.e., different university departments), including traffic management, medicine, and computer science. For our second source, we chose the translated version of the book "*Distributed Systems: Concepts and Design*", by G. F. Coulouris *et al*. [**?**], and its original version in English. The book is made up of 21 chapters, composed of over 17,550 paragraphs and 47,000 lines. To begin, we explored the contents page of the University's website, specifically for graduate studies. We found dissertation titles in both languages and ended up manually gathering the data – from both the English version and Arabic version of the website. To do this, we copied each title, in both languages, into an Excel file we created. As for our second data source, the translated book and its original English version, we proceeded to extract the content paragraph by paragraph. Thus, every paragraph was manually copied from both the Arabic and English versions of the book into an EXCEL file format (`xlsx`). Furthermore, a glossary composed of all concepts and terms from all book chapters was put together. Manual validation was performed to ensure accuracy and quality while identifying any discrepancies or errors made during sentence pairing. Linguistic specialists thoroughly analyzed our parallel data to ensure accurate alignment between the source and target sentences. They eliminated mistranslations, structural inconsistencies, and inconsistent terminology. They also evaluated the target language's fluency and naturalness while preserving the source text's intended meaning. Validation frequently necessitated iterative reviews, during which specialists rectified errors, clarified ambiguities, and standardized translation choices to uphold consistency throughout the dataset.

To gain better insight into the features of our Arabic–English parallel corpus, a statistical evaluation was performed on the aggregated dataset, shown in Figure 4. This evaluation encompasses aspects such as sentence length, character distribution, and the total size of the dataset. The findings indicated that with regards to average sentence length, it was well-balanced, as the Arabic and English sentences in the dataset contained roughly 40.1 and 43.15 words, respectively. Such a balance is beneficial for NMT models, as it guarantees that the encoder and decoder process sequences of similar lengths. Regarding statistics at the character-level, Arabic sentences demonstrated a slightly lower average of characters (242.69) in comparison to English (271.87). This disparity can be linked to the inherent morphological

and structural features of both languages, as Arabic words tend to be shorter compared to English words. Regarding sentence length, the dataset contained a broad spectrum of sentence lengths, with a maximum sentence length of 363 and 442 for Arabic and English, respectively. On the other hand, the shortest sentence length for both languages was one. Moreover, the standard deviation of sentence length is 43.45 for English and 41.34 for Arabic. The size of the dataset is notable, both in terms and words and character count, with roughly 437,650 and 470,965 words, in Arabic and English, respectively, and 2,648,718 and 2,967,244 characters in Arabic and English, respectively.

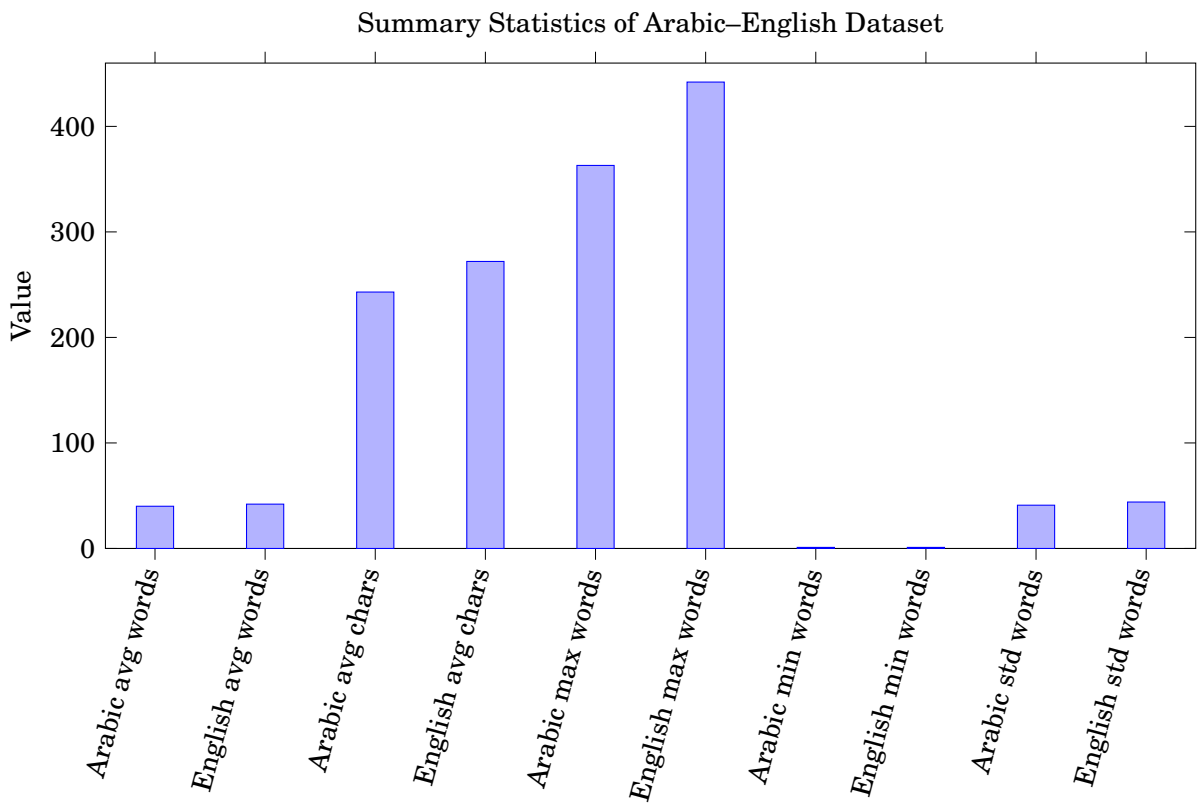

Figure 4. Statistical analysis Arabic–English parallel corpus

5.2 Preprocessing

Before the NMT model is trained, the unprocessed bilingual dataset is subjected to comprehensive preprocessing to guarantee consistency, uniformity in format, and suitable numerical representation. First, the dataset is initially loaded from a tab-separated file. In it, every line consists of a pair of parallel sentences comprised of a source sentence alongside its respective target translation. Next, every sentence passes through a preprocessing function. This function carries out several tasks including normalization, removal of noise, and handling of punctuation.

The following four preprocessing steps were applied equally to both the Arabic and English sentences in the dataset, as depicted in Figure 5.

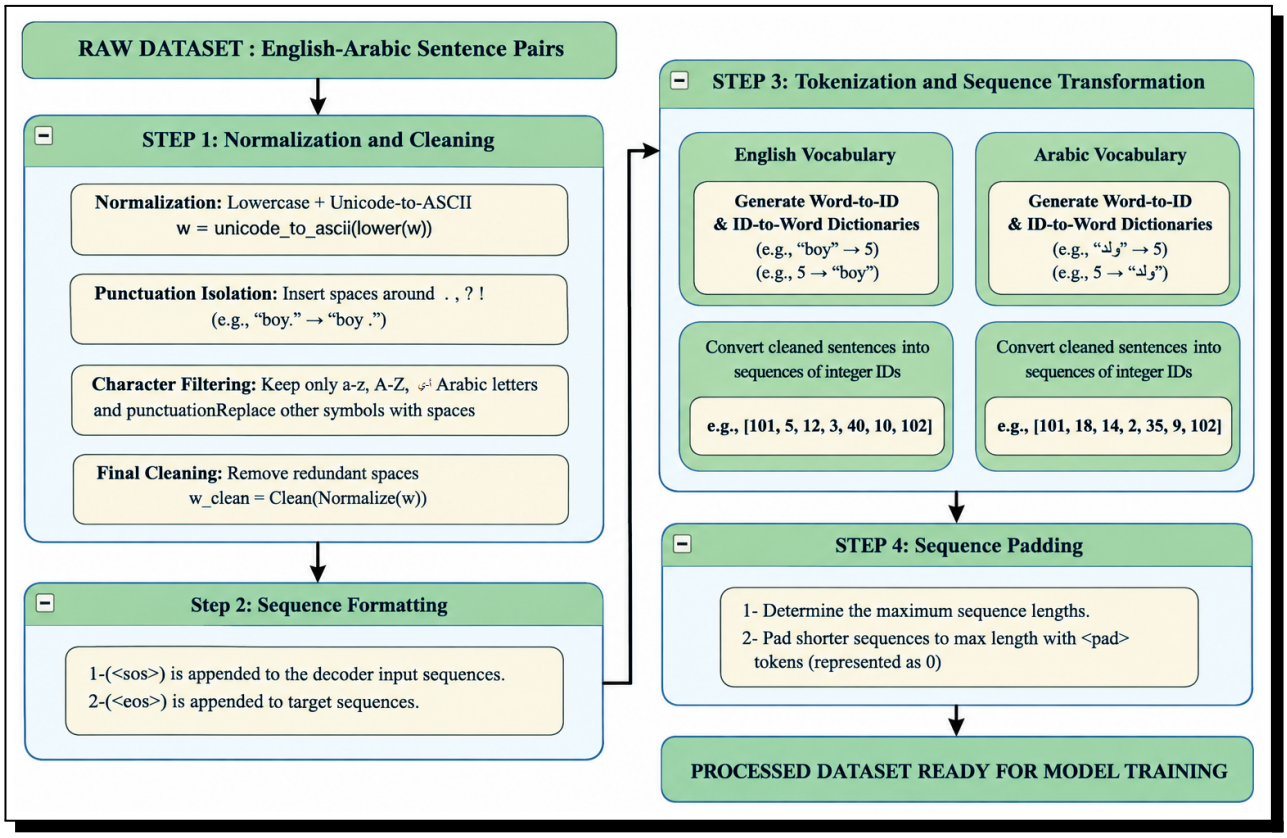

Figure 5. Text preprocessing pipeline for English and Arabic machine translation

5.2.1 Normalization and Cleaning

In the first stage of preprocessing, raw sentence pairs undergo normalization and cleaning. The purpose of which is the reduction of noise and variability. This stage covers the following:

- Conversion of English text to lowercase and stripping any leading and trailing whitespaces. Additionally, Unicode is normalized to ASCII. This removes diacritics as well as accented characters from the input text, thereby guaranteeing uniform representation of the dataset. This step in the process guarantees that visually similar characters are portrayed in a consistent way.
- Separation of punctuation marks from adjacent words – by adding spaces before and after them. Punctuation marks are now treated as independent tokens by the tokenizer, thereby enhancing alignment and learning during the model's training. For instance, the punctuation mark attached to the last word at the end of the following input sentence: 'he is a boy', would become separated from the word it is attached to with an extra space: "he is a boy.".
- Removal of multiple spaces and replacing them with a single space. This helps in maintaining uniform representation across all the dataset's input sentences.
- Filtering out unwanted characters by preserving only valid symbols (i.e., letters of the English and Arabic alphabetic (a–z, A–Z) / (–), as well as a specified set of punctuation marks). Any other symbol or character (e.g., special symbols) is substituted with spaces, thus minimizing noise within the dataset.

- Trimming the processed sentences one more time to remove any remaining whitespaces.

5.2.2 Sequence Formatting

In the sequence formatting step, sequences (both input and target) are prepared for the an encoder–decoder architecture. To this end, special tokens are added to target sentences to direct the decoding process. Target sentences are appended with an end-of-sequence token (<eos>) while decoder input sequences are prepended with a start-of-sequence token (<sos>). The purpose of the (<eos>) and (<sos>) is to signify the termination point of generation and create a right-shifted version of target sentences, respectively. This preprocessing step enhances the model's ability to learn sequence generation and termination.

5.2.3 Tokenization and Sequence Transformation

The next step involves tokenization, which utilizes a tool called the Tokenizer class by the TensorFlow Keras library to create vocabulary. To do this, each word in the corpus is designated a unique integer index. The tokenizer is applied on preprocessed source sentences in order to create a vocabulary comprised of the most frequent words, limited by the parameter MAX_VOCAB_SIZE.

A vocabulary is created for each of the dataset's two languages, English and Arabic. To do this, every unique word in the dataset is mapped to a unique integer ID, resulting in dictionaries that are word-to-ID and ID-to-word.

To ensure the preservation of vital symbols like the <sos> and <eos> tokens, tokenization is executed without any filtering. In contrast to the standard configuration, punctuation filtering is deactivated by setting the parameter filters = ''. This adjustment is essential as punctuation marks are deliberately retained during preprocessing and must be treated as tokens that have meaning. By employing these vocabularies, the cleaned sentences are converted into sequences of integer IDs, such that all words are represented by their associated numeric index. This allows for efficient numerical processing by neural networks.

This preprocessing step is illustrated in Figure 6. It demonstrates how sentences are converted into decoder input and target output sequences with special tokens, and their associated token IDs.

5.2.4 Sequence Padding

Given that neural networks require inputs to be of a fixed length, every single tokenized sequence is standardized to a consistent length using Keras' pad_sequences function. The maximum length for both the input and the output sequences is derived through empirical analysis from the dataset. Padding is placed at the end of each sequence, using <pad> tokens (denoted as 0) to maintain that all training samples have uniformity in length.

5.3 LSTM- Encoder–Decoder Architecture with Attention

The novel NMT model proposed in this research is based on LSTM encoder–decoder with attention mechanism. The model consists of five basic structures: the embedding layer, the encoder networks, the attention layer, decoder networks, and the Softmax classification layer. In Figure 7, a conventional autoregressive NMT model is shown, which utilizes the encoder–decoder architecture integrated with the attention framework.

```
Source sentence:
    لم هو هنا ؟

Target output (with <eos>):
    why is he here ? <eos>

Decoder input (with <sos>):
    <sos> why is he here ?
-------------------------------------------------------------
Word-to-ID mapping (example):

why   -> 4
is    -> 81
he    -> 15
here  -> 9
?     -> 1
<sos> -> 3
<eos> -> 2
-------------------------------------------------------------
Tokenized sequences:

Decoder input:
    [3, 4, 81, 15, 9, 1]

Target output:
    [4, 81, 15, 9, 1, 2]
```

Figure 6. Example of target sequence preparation with <sos> and <eos> tokens and corresponding tokenized representations

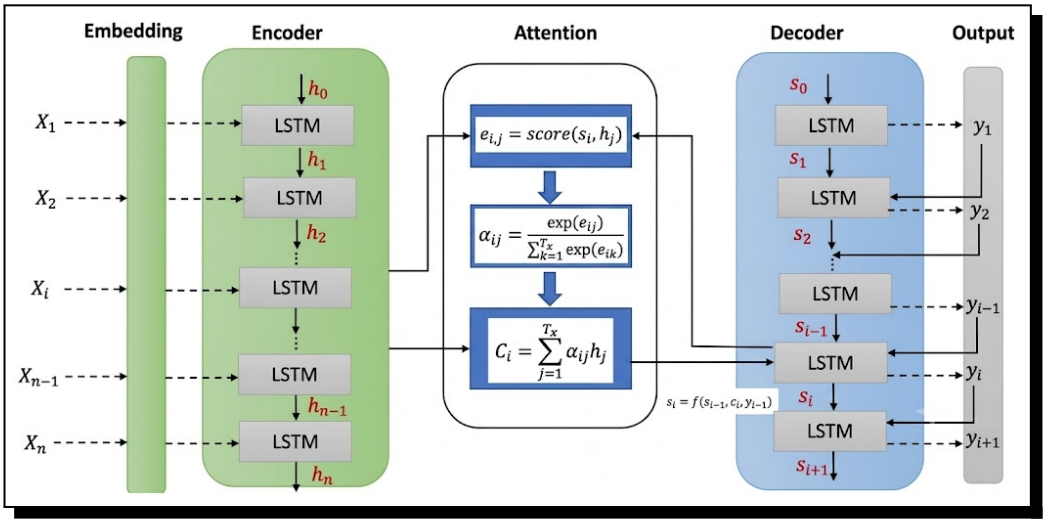

Figure 7. LSTM encoder-decoder architecture with attention mechanism

5.3.1 Embedding Layer

The first component of the model, the embedding layer, depicts the concept of continuous representation. It does this by mapping a discrete symbols x_t into a continuous vector $x_t \in R^d$, where d denotes the vector's dimension. Subsequently, these embeddings are input into subsequent layers for a more detailed feature extraction process.

5.3.2 Encoder

The second component is the encoder. This component converts input sequences into continuous vector representations that encapsulate syntactic as well as semantic information. Given an input sequence $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$, first, all of the tokens are mapped into a dense vector representation (i.e., the embedding layer). Subsequently, an LSTM network processes it in a sequential manner. In every time step t, the hidden state of the encoder is updated its hidden state in the following way:

$$h_t = \mathrm{LSTM}(x_t, h_{t-1}), \quad t = 1, 2, \ldots, n, \tag{5.1}$$

where x_t input token at time step t, h_{t-1} is the hidden state from the previous time step, and $h_t$is the current hidden state. By carrying out this iterative process, the encoder gathers contextual information incrementally while it examines the sequence from the first token to the last. Once the whole input sequence is encoded, the model produces a sequence of hidden states:

$$h = (h_1, h_2, \ldots, h_n) \tag{5.2}$$

in which every hidden state h_t contains encoded information about the input taken at position t, together with its surrounding context.

5.3.3 Attention Mechanism

The third component is the attention layer. The attention mechanism handles the constraint of encoding complete source sentences into single vectors of a fixed length. Doing this permits the decoder to concentrate on the pertinent parts of the source sentence in every step of the decoding process.

In every time step t, the attention mechanism calculates a context vector c_i as a weighted sum of encoder hidden states:

$$c_i = \sum_{j=1}^{T} \alpha_{i,j} h_j, \tag{5.3}$$

where $\alpha_{i,j}$ are attention weights that represent the relevance of the j-th source word to the current target word y_i. For every decoder time step t, the attention weights $\alpha_{i,j}$ are calculated as follows:

$$\alpha_{i,j} = \frac{\exp(e_{i,j})}{\sum_{k=1}^{T} \exp(e_{i,k})}, \tag{5.4}$$

where the alignment score $e_{i,j}$ is a measure of the relevance between the decoder state s_i and the encoder hidden state h_j, typically calculated as follows:

$$e_{i,j} = \mathrm{score}(s_i, h_j). \tag{5.5}$$

5.3.4 Decoder

The model's fourth component is the decoder layer. This layer consists of multiple LSTM units, each of which is responsible for predicting an output y_t at a time step t. To produce the hidden state (s_i) of the decoder, LSTM cells accept the previous decoder's hidden state (s_{i-1}), output from the previous time step (y_{i-1}), and the context vector (c_i) from the attention layer as:

$$s_i = f(s_{i-1}, c_i, y_{i-1}). \tag{5.6}$$

5.3.5 Softmax Layer (Output)

The fifth and final component of the proposed NMT model is the Softmax classification layer. The function of this layer is to predict the probability distribution over the target vocabulary. This layer is commonly a linear layer with Softmax activation function. It is a process that takes in inputs (i.e., hidden state representation s_t, weights associated with it, the bias as the input, and the context vector c_i) and yields a vector that contains a score for every token in the target vocabulary. Linear transformation is utilized to interpret every score as a probability distribution over the target vocabulary $|V|$ that defines an index over target tokens v_j.

5.4 Evaluation

In the last stage proposed trained NMT model evaluated on the test dataset to measure the quality of the translations by using standard evaluation metrics (e.g., BLEU score).

6. Experimental Setup

6.1 Development Environment Setup

In this study, some necessary libraries from TensorFlow and Keras are imported to assist the development of the NMT.

- *Tokenizer from* (`tensorflow.keras.preprocessing.text.Tokenizer`) is used for text preprocessing by converting text into sequences of integers, where each word or token in the text is mapped to a unique integer based on its frequency in the dataset.
- `pad_sequences` (`tensorflow.keras.preprocessing.sequence.pad_sequences`) is used to ensure that all input and output sequences are of the same length. Since sentences vary in length, padding is applied (typically with zeros) .
- *Keras Backend* (`tensorflow.keras.backend` as K) provides low-level backend functions that support tensor operations and model manipulation. It is useful for custom operations and controlling the computational graph during model development.

6.2 Evaluation Metrics

To evaluate the proposed model, we have used *The Bilingual Evaluation Understudy* (BLEU) (Papineni *et al*. [24]) score is a metric that is widely used to assess the quality of machine translation. It assesses the fluency and adequacy of the text for n-grams of size 1 through 4. It does this by focusing on three main aspects: n-gram precision, *Brevity Penalty* (BP) and a clipping mechanism. n-gram precision is used for comparing the generated and reference translations. BP is used to adjust for situations where the generated sentence length is significantly shorter than the reference. As for the clipping mechanism, it is used to avoid overestimating the matches by counting repeated n-grams multiple times, limiting the count to the maximum found in the reference. To find precision, the number of n-grams that were matched is divided by the total number of n-grams. The precision p_n is calculated as follows:

$$p_n = \frac{\sum_{p\in\text{hypothesis}} n\text{-gram} \sum_p \text{Count}_{\text{clip}}(n\text{-gram})}{\sum_{p\in\text{hypothesis}} n\text{-gram} \sum_p \text{Cont}(n\text{-gram})}. \tag{6.1}$$

BLEU counts the highest frequency of n-gram matches. To prevent the same word (n-gram) from being counted multiple times, BLEU restricts the match count for each n-gram to

the maximum count observed in any single reference. The formula for calculating a clipped count is defined as:

$$\text{Count}_{\text{clip}} = \min\left(n\text{-gram count}, \max_{r\in R}(n\text{-gram count in } r)\right). \tag{6.2}$$

BP is applied when the length of the generated translation is less than that of the reference, to reduce the influence of sentence length on the BLEU score. The formulas for BP and BLEU are presented as follows:

$$\text{BP} = \begin{cases} 1, & \text{if } |h| > |r|, \\ e^{(1-|r|/|h|)}, & \text{if } |h| \le |r|, \end{cases} \tag{6.3}$$

$$\text{BLEU} = \text{BP} \cdot \exp\left(\sum_{n=1}^{N} w_n \log p_n\right). \tag{6.4}$$

7. Experimental Results

In this section, we have comprehensive descriptions of training and testing our model with the multi-domain Arabic-English parallel corpus and experimental results with evaluation results are presented. The performance of the proposed model is evaluated using both training metrics (loss and token-level accuracy) and a test-evaluation metric (BLEU score). These assessments give a complete understanding of the model's behavior of learning and quality of translation.

7.1 Training Loss and Accuracy Analysis

We train the model using the Adam optimizer with a learning rate of 0.002 for 30 epoch with a batch size of 8. For four domains (Legal, Computer Science, Science, and Medical) in both Arabic to English and English to Arabic translations. The training loss is computed using the masked *Sparse Categorical Cross-Entropy function*, which measures the difference between the predicted token probability distributions and the actual target tokens while ignoring padding positions. The loss is defined as the negative logarithm of the predicted probability assigned to the correct word as follows:

$$\mathcal{L} = -\log(P_y), \tag{7.1}$$

where $P_y$is the model's predicted probability for assigned to the correct word. The total loss is computed as the average of the individual losses across all words as:

$$\text{Total Loss} = -\frac{1}{n}\sum_{i=1}^{n} \log(P_{y_i}). \tag{7.2}$$

Token-level accuracy is used to evaluate the correctness of predicted tokens during training, excluding padding tokens using a masking mechanism and it's calculated as follows:

$$\text{Accuracy} = \frac{\text{correct}}{\text{total real}}. \tag{7.3}$$

Table 1 and Table 2 explains the output results when we train model during translation from Arabic to English and from English to Arabic respectively. Through the epoch the training results show a clear improvement in model performance. Firstly, the model begins with a high loss and very low accuracy, suggesting weak predictions. As training progresses, the loss steadily decreases while accuracy increases, indicating successful learning. By the final epoch, the model achieves a low loss and high accuracy.

Table 1. Output results when we train and validate samples during translation from Arabic to English

Epoch	Loss	Accuracy
1	2.2743	0.0000
2	1.3714	0.1509
3	1.8011	0.1642
⋮	⋮	⋮
30	0.0475	0.9565

Table 2. Output results when we train and validate samples during translation from English to Arabic

Epoch	Loss	Accuracy
1	1.8951	0.0010
2	1.7483	0.1296
3	0.9562	0.2353
⋮	⋮	⋮
30	0.0461	0.9630

7.2 Testing Result

Our NMT was assessed across four distinct domains (Legal, Computer Science, Science and Medical), where we calculated BLEU scores for both translation paths: Arabic to English (Ar → En) and English to Arabic (En → Ar). Table 3 presents a summary of the test BLEU scores obtained under the training hyperparameter for each domain in both translation directions.

Table 3. Test BLEU scores of proposed NMT for each domain in both direction (Arabic to English and English to Arabic)

Domain	Arabic → English	English → Arabic
Legal	44.30	48.67
Computer Science	43.29	45.31
Science	29.86	28.66
Medical	40.13	42.85

Discussion

Figure 8 illustrates the BLEU score distribution across different domains for both translation directions (Arabic → English and English → Arabic). The figure clearly highlights the variation in translation quality depending on the domain.

Overall, the model achieves its best performance in the *Legal domain*, with BLEU scores of *44.30* (Ar → En) and *48.67* (En → Ar). This indicates that the model handles this domain relatively well, likely due to more consistent vocabulary, simpler sentence structures, or better representation in the training data. The higher score in the English → Arabic direction suggests that the model may be slightly more effective in generating Arabic text for this domain.

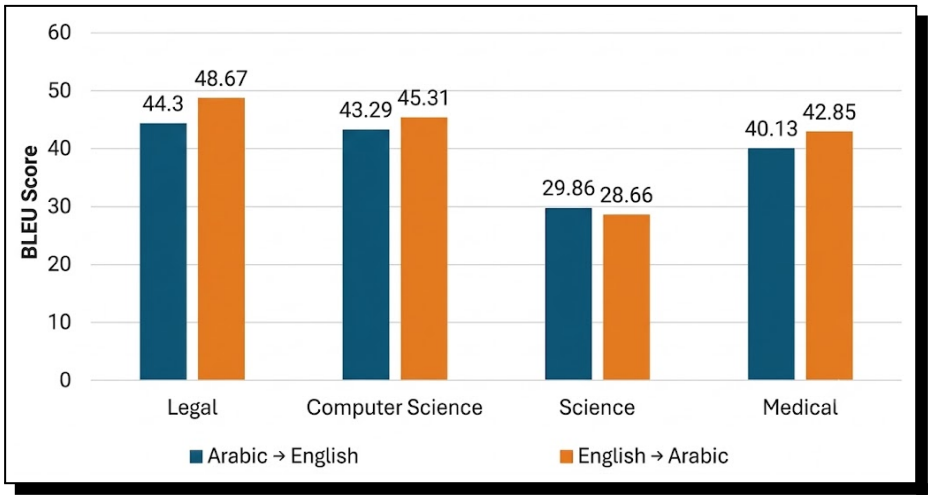

Figure 8. BLEU score distribution across different domains for both translation directions (Arabic → English and English → Arabic)

In the *Computer Science domain*, the performance remains relatively strong, with BLEU scores of *43.29* (Ar → En) and *45.31* (En → Ar). Although slightly lower than the Legal domain, the results still indicate stable translation quality. This suggests that the model generalizes reasonably well to this domain, possibly due to moderate linguistic complexity and domain similarity with the training data.

However, a significant drop in performance is observed in the *Science domain*, where BLEU scores decrease to *29.86* (Ar → En) and *28.66* (En → Ar). This indicates that the model struggles considerably with this domain. The lower scores may be attributed to factors such as domain-specific terminology and higher syntactic variability. The nearly symmetric low performance in both directions suggests a general difficulty rather than direction-specific issues.

For the *Medical domain*, the model achieves moderate performance, with BLEU scores of *40.13* (Ar → En) and *42.85* (En → Ar). These results fall between the high-performing (Legal, Computer Science) and low-performing (Science) domains, indicating that the model can partially adapt to this domain but still faces challenges, possibly due to mixed linguistic patterns or partial domain mismatch.

Across all domains, a consistent pattern can be observed where *English → Arabic translation slightly outperforms Arabic → English*.

7.3 Comparison

We assess the BLEU scores of our proposed NMT model against:

- The OPUS-MT model available on Hugging Face ([Arabic to English] (https://huggingface.co/Helsinki-NLP/opus-mt-ar-en) and [English to Arabic] (https://huggingface.co/Helsinki-NLP/opus-mt-en-ar)).
- Google Translate as evaluated through its API using identical test sets.

Table 4 illustrates the BLEU scores for Arabic to English translations across different domains using our proposed NMT model alongside the OPUS (ar-en) MT system and Google Translate. The results demonstrate that our model consistently surpasses all other models across every domain evaluated.

Table 4. BLEU scores across domains for proposed NMT model, OPUS (ar-en), and Google Translate for Arabic → English direction

Domain	Proposed Model	OPUS (ar-en)	Google Translate
Legal	44.30	21.40	24.05
Computer Science	43.29	17.20	22.21
Science	29.86	15.58	13.53
Medical	40.13	28.78	25.37

Observations. In order to evaluate the effectiveness of our proposed NMT model, we conducted a comparative analysis against the OPUS MT system and Google Translate across four distinct domains, utilizing BLEU as the metric for assessment. As illustrated in Figure 9, our model consistently demonstrates superior performance over both OPUS MT and Google Translate in Arabic-to-English translations across all domains examined. Specifically, it exceeds the performance of the OPUS MT system by an average margin of +18.66 BLEU points and outperforms Google Translate by approximately +18.11 BLEU points.

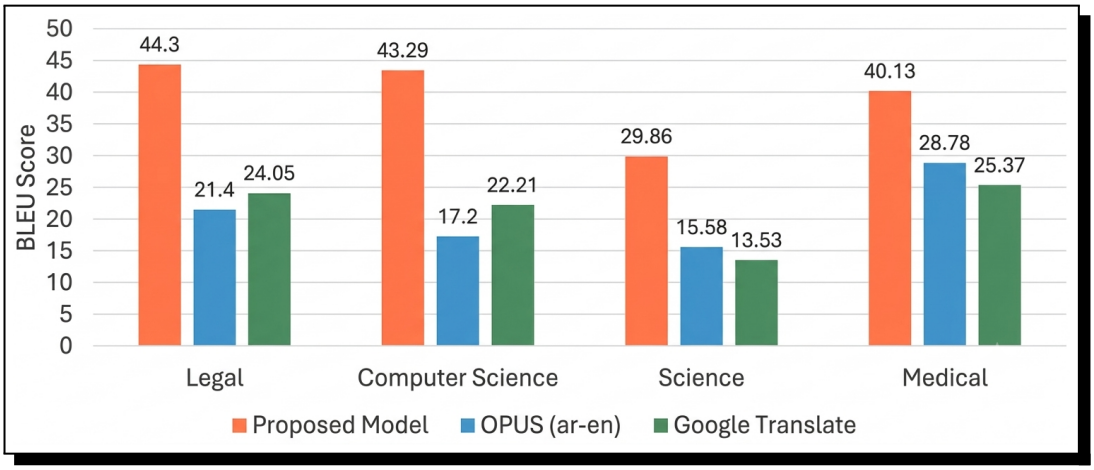

Figure 9. BLEU scores across domains for proposed NMT model, OPUS (ar-en), and Google Translate for Arabic → English direction

For example, in the Legal domain, our model achieves a 1BLEU score of 44.30, which significantly surpasses those of OPUS MT (21.40) and Google Translate (24.05).

Similarly, within the Computer Science and Medical domains, our model secures scores of 43.29 and 40.13 respectively; in contrast, OPUS MT records significantly lower scores of 17.20 and 28.78 while Google Translate achieves scores of 22.21 and 25.37.

Even in the more challenging Scientific domain, our model exhibits a noteworthy advantage with a score of 29.86 compared to OPUS MT's 15.58 and Google's 13.53. This indicates that even in more challenging or low-resource domains, the proposed approach remains more robust than the baselines.

Overall, Google Translate generally performs better than OPUS in most domains, except in the third domain where OPUS slightly surpasses it. However, both systems fall significantly behind the proposed model. This confirms that general-purpose translation systems are less effective when applied to specialized or domain-specific data.

In summary, the results demonstrate that the proposed model delivers substantial and consistent improvements over widely used baseline systems, highlighting its effectiveness in domain adaptation and its suitability for specialized translation tasks.

Similarly, Table 5 shows the BLEU scores for English → Arabic translation, as with the Arabic →English direction, the proposed NMT model superior performance, highlighting its effectiveness in both translation directions.

Table 5. BLEU scores across domains for proposed model, OPUS (en-ar), and Google Translate for English → Arabic direction

Domain	Proposed Model	OPUS (en-ar)	Google Translate
Legal	48.67	24.37	33.27
Computer Science	45.31	18.81	23.06
Science	28.66	10.58	20.71
Medical	42.85	22.18	27.03

Observations. Beyond the Arabic → English translation, we assessed our NMT model in the English → Arabic context across four identical domains, comparing its performance to that of an OPUS MT system and Google Translate. The findings illustrated in Figure10 indicate a consistent advantage for our model. Specifically, it achieves average BLEU score improvements of +22.39 points over OPUS (en-ar) and +15.36 points over Google Translate. For instance, in the Legal domain, our model records a BLEU score of 48.67, markedly surpassing OPUS (en-ar) at 24.37 and Google Translate at 33.27. Additionally, in the Computer Science domain, our model maintains a strong lead with a BLEU score of 45.31; meanwhile, OPUS MT and Google Translate obtain scores of 18.81 and 23.06, respectively. In the Medical fields, it registers scores of 42.85, compared to OPUS (en-ar) scores of 22.18 and Google's scores of 27.03. Even in more challenging areas such as Science, our model demonstrates significant enhancements with a BLEU score of 28.66 compared to OPUS (en-ar) at 10.58 and Google at 20.71. These outcomes corroborate that the proposed model not only improves translation quality from Arabic to English but also offers considerable advancements in the opposite direction across various domains.

8. Conclusion

While there exist several MT tools and models designed for translation of Arabic text, the translation quality is often lacking, particularly with texts that fall out-of-domain. A major technical difficulty combined with AMT restricts the scarcity of bilingual datasets for such texts that are out-of-domain. Therefore, this paper aimed at enhancing AMT and improving translation quality across four specific domains: Legal, Computer Science, Science and Medical. To achieve this, we designed a multi-domain parallel corpus specifically for *Arabic-English* (AR-EN) translation, which was later utilized to train the bilingual Arabic-English NMT model. The proposed NMT model uses the encoder–decoder architecture that employs *long short-term memory* (LSTM) networks, which are themselves integrated with attention mechanisms. To further evaluate the effectiveness of our proposed model, we conducted a comparative analysis with the OPUS MT system and Google Translate across these domains using BLEU

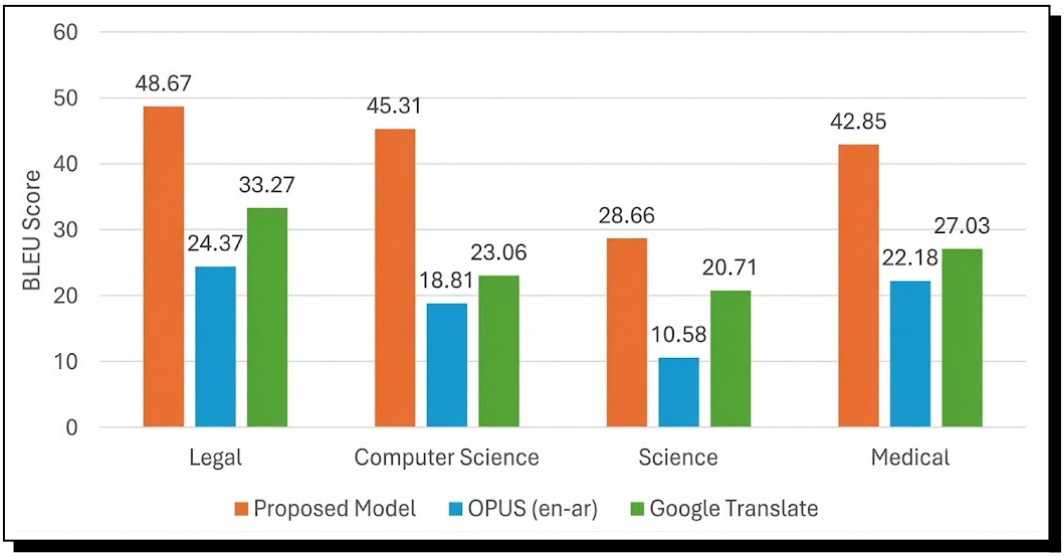

Figure 10. BLEU scores across domains for proposed model, OPUS (en-ar), and Google Translate for English → Arabic direction

as the evaluation metric. In the direction from Arabic to English, our model consistently outperformed both OPUS MT and Google Translate across all domains. Specifically, it exceeded OPUS MT by an average of +18.66 BLEU points and surpassed Google Translate by approximately +18.11 BLEU points. Similarly, for the English to Arabic translation direction, our results consistently indicated that our model is superior. It achieved average BLEU gains of +22.39 points over OPUS MT and +15.36 points over Google Translate.

Competing Interests

The authors declare that they have no competing interests.

Authors' Contributions

All the authors contributed significantly in writing this article. The authors read and approved the final manuscript.

References

[1] N. Abdullah, Transfer-based Arabic to English noun sentence translation using shallow segmentation, *The International Arab Journal of Information Technology* **15**(5) (2018), 889 – 893.

[2] E. A. Alghamdi, J. Zakraoui and F. A. Abanmy, Domain adaptation for Arabic machine translation: Financial texts as a case study, *Applied Sciences* **14**(16) (2024), 7088, DOI: 10.3390/app14167088.

[3] A. Almahairi, K. Cho, N. Habash and A. C. Courville, First result on Arabic neural machine translation, *arXiv:1606.02680* (2016), 6 pages, DOI: 10.48550/arXiv.1606.02680.

[4] A. T. Al-Taani and E. M. Hailat, A direct English-Arabic machine translation system, *Information Technology Journal* **4**(3) (2005), 256 – 261, DOI: 10.3923/itj.2005.256.261.

[5] W. Aransa, *Statistical Machine Translation of the Arabic Language*, Ph.D. dissertation, LIUM - Laboratoire d'Informatique de l'Université du Maine, (2015), URL: https://theses.hal.science/tel-01316544v1.

[6] D. Ataman, A. Birch, N. Habash, M. Federico, P. Koehn and K. Cho, Machine translation in the era of large language models: A survey of historical and emerging problems, *Information* **16**(9) (2025), 723, DOI: 10.3390/info16090723.

[7] D. Bahdanau, K. Cho and Y. Bengio, Neural machine translation by jointly learning to align and translate, in: *Proceedings of the International Conference on Learning Representations* (ICLR'2015, May 7-9, 2015), 15 pages, (2015).

[8] S. Berrichi and A. Mazroui, Guiding word alignment with prior knowledge to improve English-Arabic machine translation, in: *BDIoT'19: Proceedings of the 4th International Conference on Big Data and Internet of Things*, Article number 9 (2019), pp. 1 – 5, DOI: 10.1145/3372938.3372957.

[9] M. R. Costa-Jussà, M. Farrús, J. B. Mariño and J. A. R. Fonollosa, Study and comparison of rule-based and statistical Catalan-Spanish machine translation systems, *Computing and Informatics* **31**(2) (2012), 245 – 270, URL: https://repositori.upf.edu/server/api/core/bitstreams/8951cb08-54b8-44f9-a0ec-6aa892042218/content.

[10] G. F. Coulouris, J. Dollimore, T. Kindberg and G. Blair, *Distributed Systems: Concepts and Design*, 5th edition, Addison-Wesely, Boston, xvi + 1047 pages (2012).

[11] H. Daumé III and J. Jagarlamudi, Domain adaptation for machine translation by mining unseen words, in: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, Association for Computational Linguistics, pp. 407 – 412, (2011), URL: https://aclanthology.org/P11-2071/.

[12] R. Ehab, English-Arabic hybrid machine translation system using EBMT and translation memory, *International Journal of Advanced Computer Science and Applications* **10**(1) (2019), DOI: 10.14569/IJACSA.2019.0100126.

[13] R. Ehab, E. Amer and M. Gadallah, Example-based English to Arabic machine translation: Matching stage using internal medicine publications, in: *ICSIE'18: Proceedings of the 7th International Conference on Software and Information Engineering*, (2018), pp. 131 – 135, DOI: 10.1145/3220267.3220294.

[14] F. Z. El Idrysy, S. Hourri, I. El Miqdadi, A. Hayati, Y. Namir, B. Ncir and J. Kharroubi, Unlocking the language barrier: A journey through Arabic machine translation, *Multimedia Tools and Applications* **84**(4) (2025), 14071 – 14104, DOI: 10.1007/s11042-024-19551-8.

[15] A. Farhat and A. Al-Taani, A rule-based English to Arabic machine translation approach, in: *Proceedings of the The International Arab conference on information technology (ACIT'2015)*, (2015), 7 pages.

[16] 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, Cairo, Egypt, 2022), pp. 378 – 383, (2022), DOI: 10.1109/MIUCC55081.2022.9781776.

[17] I. Gashaw and H. L. Shashirekha, Amharic-Arabic neural machine translation, in: *Proceedings of the 5th International Conference on Data Mining and Applications* (DMAP2019, December 14, 2019), 15 pages.

[18] H. Hamed, A. M. Helmy and A. Mohammed, Deep learning approach for translating Arabic holy Quran into Italian language, in *2021 International Mobile, Intelligent, and Ubiquitous Computing Conference* (MIUCC, Cairo, Egypt, 2021), pp. 193 – 199 (2021), DOI: 10.1109/MIUCC52538.2021.9447650.

[19] I. T. Khemakhem, S. Jamoussi and A. B. Hamadou, Improving English-Arabic statistical machine translation with morpho-syntactic and semantic word class, *International Journal of Intelligent Systems Technologies and Applications* **19**(2) (2020), 172 – 190, DOI: 10.1504/IJISTA.2020.107225.

[20] P. Koehn and R. Knowles, Six challenges for neural machine translation, in: *Proceedings of the First Workshop on Neural Machine Translation*, pp. 28 – 39, Vancouver, Association for Computational Linguistics, (2017), URL: https://aclanthology.org/W17-3204/.

[21] M.-T. Luong, H. Pham and C. D. Manning, Effective approaches to attention-based neural machine translation, in: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing* (Lisbon, Portugal, September 17-21, 2015), L. Màrquez, C. Callison-Burch and J. Su (editors), Association for Computational Linguistics, pp. 1412 – 1421, (2015), DOI: 10.18653/v1/D15-1166.

[22] D. Moussallem, M. Wauer and A.-C. N. Ngomo, Machine translation using semantic web technologies: A survey, *Journal of Web Semantics* **51** (2018), 1 – 19, DOI: 10.1016/j.websem.2018.07.001.

[23] Noor-e-Hira, S. A. Rauf, K. Kiani, A. Zafar and R. Nawaz, Exploring transfer learning and domain data selection for the biomedical translation, in: *Proceedings of the Fourth Conference on Machine Translation* (WMT, Vol. 3: Shared Task Papers (Day 2), Florence, Italy, August 1-2, 2019), Association for Computational Linguistics, pp. 156 – 163, (2019), DOI: 10.18653/v1/W19-5419.

[24] 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 on Association for Computational Linguistics* (Philadelphia, Pennsylvania, 2002), pp. 311 – 318 (2002), DOI: 10.3115/1073083.1073135.

[25] L. Qun and Z. Xiaojun, Machine translation: General, in: *Routledge Encyclopedia of Translation Technology*, 1st edition, C. Sin-Wai (editor), pp. 105–119, Routledge (2014).

[26] Y. Shalhoub-Awwad and M. Leikin, The lexical status of the root in processing morphologically complex words in Arabic, *Scientific Studies of Reading* **20**(4) (2016), 296 – 310, DOI: 10.1080/10888438.2016.1180525.

[27] S. P. Singh, A. Kumar, H. Darbari, L. Singh, A. Rastogi and S. Jain, Machine translation using deep learning: An overview, in: *2017 international conference on computer, communications and electronics* (Comptelix, Jaipur, India, 2017), pp. 162 – 167, (2017), DOI: 10.1109/COMPTELIX.2017.8003957.

[28] I. Sutskever, O. Vinyals and Q. V. Le, Sequence to sequence learning with neural networks, in: *Proceedings of the 28th International Conference on Neural Information Processing Systems*, Vol. 2 (Series NIPS'14, Cambridge, MA), MIT Press, USA, pp. 3104 – 3112, (2014).

[29] G. Thurmair, Comparing rule-based and statistical MT output, in: *Proceedings of the Workshop on the Amazing Utility of Parallel and Comparable Corpora* (Lisbon, Portugal, 2004), pp. 5 – 9, (2004), URL: https://aclanthology.org/www.mt-archive.info/LREC-2004-Thurmair-2.pdf.

[30] R. K. Yadav, S. Harwani, S. K. Maurya and S. Kumar, Intelligent chatbot using GNMT, SEQ-2-SEQ techniques, in: *2021 International Conference on Intelligent Technologies* (CONIT, Hubli, India, 2021), pp. 1 – 5, (2021), DOI: 10.1109/CONIT51480.2021.9498485.

[31] J. Zakraoui, M. Saleh, S. Al-Maadeed and J. M. AlJa'am, Evaluation of Arabic to English machine translation systems, in: *2020 11th International Conference on Information and Communication Systems* (ICICS, Irbid, Jordan, 2020), pp. 185 – 190, (2020), DOI: 10.1109/ICICS49469.2020.239518.