Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

An Optimized Approach for Handling Limited Vocabulary Constraints and Word Order Divergence in English-Urdu Neural Machine Translation

Rabail Asghar
COMSATS University Islamabad
Junaid Akram
junaidakram@cuilahore.edu.pk
COMSATS University Islamabad
Yella Mehroze
COMSATS University Islamabad
Zaheer Ahmad Gondal
Huazhong University of Science and Technology

Research Article

Keywords: Neural Machine Translation (NMT), Natural Language Processing (NLP), English-Urdu Machine Translation (MT), Transformer, Out-of-Vocabulary (OOV) words

Posted Date: November 5th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-9389246/v2

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

An Optimized Approach for Handling Limited Vocabulary Constraints and Word Order Divergence in English-Urdu Neural Machine Translation

Rabail Asghar[1], Junaid Akram[1*], Yella Mehroze[2], Zaheer Ahmad Gondal[3]

[1]Department of Computer Science, COMSATS University Islamabad, Lahore, Pakistan.
[2]Department of Computer Science, COMSATS University Islamabad, Lahore, Pakistan.
[3]School of Computer Science, Huazhong University of Science and Technology, China.

*Corresponding author(s). E-mail(s): junaidakram@cuilahore.edu.pk;

Abstract

Machine Translation (MT) in Natural Language Processing (NLP) seeks to bridge language barriers by developing automated systems capable of translating human languages. Transformer-based Neural Machine Translation (NMT) has surpassed traditional approaches in translation quality; however, low-resource and morphologically rich language pairs such as English–Urdu remain challenging due to vocabulary sparsity and syntactic word-order divergence. This study presents a framework that addresses these challenges through systematic evaluation of subword tokenization strategies and Dynamic Position Encoding (DPE) based word-order modeling. A large-scale English–Urdu parallel corpus was compiled, cleaned, and publicly released to support reproducible research. Multiple tokenization approaches, including Byte-Pair Encoding (BPE) and SentencePiece variants, were evaluated using Transformer based translation models, and the best-performing tokenizer–model combinations were further optimized by tuning the reordering loss weight. Experimental results showed that Subword-NMT BPE achieved the strongest performance for English–Urdu translation, while SentencePiece-BPE performed best for Urdu–English translation. Furthermore, the optimized Transformer variant consistently outperformed the Transformer Base baseline in both translation directions. These findings demonstrate the effectiveness of jointly addressing vocabulary sparsity and word-order divergence for improving English–Urdu neural machine translation.

Keywords: Neural Machine Translation (NMT), Natural Language Processing (NLP), English-Urdu Machine Translation (MT), Transformer, Out-of-Vocabulary (OOV) words

1 Introduction

Machine Translation (MT) aims to automatically translate text between human languages and remains a fundamental task in Natural Language Processing (NLP). As one of the longest-standing research areas in NLP, MT integrates concepts from linguistics, statistics, mathematics, and artificial intelligence to facilitate communication across language barriers (Jindal et al., 2018). The growing need for cross-lingual communication and access to multilingual information has significantly increased the demand for accurate and efficient translation systems, making MT an essential technology for improving access to information and facilitating communication across

diverse linguistic communities (Andrabi et al., 2022; Craciunescu et al., 2004). Over the past decades, MT has evolved from rule-based and statistical approaches to Neural Machine Translation (NMT), with Transformer architectures (Vaswani et al., 2017) emerging as the dominant paradigm due to their ability to effectively model long-range dependencies through self-attention mechanisms. Urdu is the tenth most spoken language in the world, with approximately 246 million speakers (Eberhard et al., 2026). Given the large volume of English-language content, effective English–Urdu translation is essential for improving information accessibility (Choudhary et al., 2020). Despite its significance, English–Urdu translation remains relatively underexplored in NMT research.

Several challenges complicate English–Urdu translation. A major issue is vocabulary sparsity, which gives rise to Out-of-Vocabulary (OOV) words such as named entities, domain-specific terms, and neologisms (Luong et al., 2015). Another challenge stems from the syntactic divergence between English (Subject–Verb–Object, SVO) and Urdu (Subject–Object–Verb, SOV), resulting in structural differences that complicate machine translation (Khan et al., 2018). These factors can negatively affect translation fluency and adequacy.

Despite recent advances, handling vocabulary sparsity and English–Urdu word-order divergence remains challenging. This study investigates these issues through experimental evaluation of subword tokenization strategies and transformer based models.

The main contributions of this study are:

1. Collected and unified English–Urdu parallel corpora, applying a research-informed preprocessing pipeline to create a high-quality, standardized translation dataset, which has been publicly released on Mendeley Data to facilitate reproducibility and support further research.
2. Investigated multiple subword tokenization strategies including Byte-Pair Encoding (BPE) and SentencePiece variants to reduce vocabulary sparsity and effectively handle OOV words.
3. Evaluated a Dynamic Position Encoding (DPE) Transformer architecture and optimized its reordering loss weight to better address syntactic divergence between English and Urdu.
4. Identified the optimal tokenizer–model combination through systematic evaluation and performed performance analysis, demonstrating improved translation quality.

2 Related Work

English–Urdu machine translation has evolved through several paradigms, ranging from rule-based and statistical methods to neural, multilingual, and transfer-learning approaches. This section reviews these developments and highlights the limitations that motivate this study.

2.1 Rule-Based and Statistical Machine Translation

Early English–Urdu machine translation systems primarily relied on rule-based approaches that exploited the close structural relationship between Hindi and Urdu. Mahesh and Sinha (2009) proposed an English–Urdu translation framework derived from an existing English–Hindi MT system through lexical mapping and grammatical transformations. Later, Khan et al. (2018) investigated translation divergence between English (SVO) and Urdu (SOV), proposing computational algorithms for the automatic detection and handling of structural, lexical, categorial, and thematic divergences. While these approaches explicitly modeled linguistic phenomena, they depended heavily on handcrafted rules and manually engineered language resources.

With the emergence of parallel corpora, phrase-based SMT gradually replaced rule-based methods. Ali et al. (2013) developed an English–Urdu SMT system using the Moses framework (Koehn et al., 2007) and highlighted challenges arising from limited parallel data and linguistic differences between the two languages. Several studies subsequently addressed word-order divergence through explicit reordering techniques. Jawaid and Zeman (2011) proposed a syntax based preordering framework that transforms English parse trees into structures more closely aligned with Urdu word order. Similarly, Visweswariah et al. (2011) introduced a parser-independent statistical reordering model

that learns source-side word-order transformations directly from aligned bilingual data.

Despite these advances, SMT systems remained sensitive to data sparsity and long-distance reordering errors, motivating the transition toward neural approaches.

2.2 Neural Machine Translation Approaches

Early NMT systems employed recurrent encoder–decoder architectures with attention mechanisms to learn source–target mappings in an end-to-end manner. Saini and Sahula (2020) explored attention-based Bi-LSTM architectures across several Indian languages, including Urdu, demonstrating the applicability of neural approaches under limited-resource conditions.

Subsequent research investigated alternative neural architectures to improve translation quality and sequence modeling. Rai et al. (2020) proposed the Convolutional English-to-Urdu Translation (CEUT) model, a convolutional sequence-to-sequence framework designed to improve source–target attention mapping in English–Urdu translation. More recently, Israr et al. (2024) introduced a fully convolutional neural network encoder for Urdu–English NMT to improve long-range dependency modeling. Recent studies have continued to enhance neural architectures through complementary strategies. Israr et al. (2025) incorporated external knowledge into a multi-encoder attention framework for Urdu–English translation, while Zuhra et al. (2026) improved Transformer-based English–Urdu translation by integrating averaged word embeddings enriched with semantic and morphological information. Researchers have also explored the incorporation of linguistic knowledge into neural architectures. Ul Hassan et al. (2024) proposed a Linguistics Knowledge-Driven Multi-Task (LKMT) framework that integrates part-of-speech and dependency information into a Transformer-based NMT model through multi-task learning.

Although these studies enhanced translation quality through architectural improvements, they generally addressed English–Urdu translation at the model level, with comparatively less attention given to vocabulary segmentation and word-order divergence.

2.3 Multilingual and Transfer Learning Strategies

To address data scarcity in low-resource machine translation, researchers have increasingly explored multilingual and transfer learning approaches. Ko et al. (2021) proposed the NMT-Adapt framework, which combines adversarial adaptation, denoising autoencoding, and back-translation to transfer translation knowledge from high-resource languages to related low-resource languages using monolingual data. Experiments involving Hindi and Urdu demonstrated the potential of exploiting linguistic overlap between related languages in resource-constrained settings. More recently, Bala Das et al. (2023) developed a Transformer-based multilingual NMT system covering 15 Indic languages (including Urdu) and English. The study further explored the use of language relationships, back-translation, and domain adaptation to improve translation quality for low-resource languages. Wagh et al. (2025) proposed a multilingual transfer learning framework for English–Assamese, English-Bodo, and English-Manipuri translation using Hindi and Bengali as high-resource source languages. Although Urdu was not included, multilingual pre-training and fine-tuning significantly improved translation quality for low-resource Indic languages. More recently, Luo et al. (2026) introduced a large-scale multilingual machine translation framework supporting 60 languages, including Urdu. The framework employed multilingual fine-tuning and prompting strategies to enhance cross-lingual knowledge transfer across diverse translation directions.

Together, these studies demonstrate that multilingual learning and cross-lingual transfer can substantially improve translation quality in low-resource settings. However, the effectiveness of such approaches varies across language pairs and depends on the extent to which knowledge can be shared between languages.

Collectively, previous research has advanced English–Urdu machine translation through different approaches. Rule-based and SMT systems relied heavily on handcrafted resources and explicit reordering techniques, while neural approaches improved contextual modeling through increasingly sophisticated architectures.

Multilingual and transfer learning methods further alleviated data scarcity by leveraging related languages and cross-lingual knowledge transfer. Nevertheless, comprehensive comparative evaluation of subword tokenization strategies to address vocabulary and explicit modeling of English–Urdu word-order divergence remain comparatively underexplored. To address mentioned gaps, this study evaluates multiple subword tokenization methods and uses dynamic position encoding based transformer framework with a publicly released and reproducibly processed large-scale parallel corpus.

3 Methodology

This section describes the methodology adopted to enhance NMT for the English–Urdu language pair. Workflow 1 illustrates the proposed pipeline, beginning with the preparation of monolingual corpora for tokenizer training and a parallel corpus for translation. The trained tokenizers are subsequently evaluated through Transformer-based translation experiments, followed by hyperparameter tuning and comparative analysis. Each stage of the proposed approach is described in detail in the following subsections.

3.1 Dataset Collection and Preprocessing for Tokenizer Training

To improve vocabulary coverage for English–Urdu Neural Machine Translation (NMT), high-quality monolingual data is essential for training effective tokenizers. Therefore, a diverse English–Urdu corpus was compiled from multiple domains, including religious texts (Quran, Bible), history, news, science, articles, poetry, and movie reviews[1][2]. This broad coverage ensures tokenizers learn varied linguistic patterns relevant to translation.

The collected corpus underwent thorough preprocessing to ensure consistency and quality. Symbolic content such as hashtags, URLs, and emojis was removed. White-space normalization involved trimming leading/trailing spaces and collapsing consecutive spaces, while preserving meaningful internal spacing. Empty lines were filtered out to retain only valid text content.

[1] https://github.com/urduhack/awesome-urdu

[2] https://www.kaggle.com/datasets/jkkphys/english-wikipedia-articles-20170820-sqlite

Workflow 1 Optimized Translation Workflow for English-Urdu Language Pair

Require: $\text{EN}_{\text{corpus}}$ (English Corpus), $\text{UR}_{\text{corpus}}$ (Urdu Corpus) for tokenizer training, $\text{EN-UR}_{\text{parallel}}$ (English–Urdu Parallel Corpus) for Transformer training

Ensure: $ParameterOptimizedModel$: Best-trained Transformer model for English–Urdu translation

$\text{EN}_{\text{processed}} \leftarrow preprocess(\text{EN}_{\text{corpus}})$
$\text{UR}_{\text{processed}} \leftarrow preprocess(\text{UR}_{\text{corpus}})$
$\text{EN-UR}_{\text{processed parallel}} \leftarrow preprocess(\text{EN-UR}_{\text{parallel}})$
for $i \leftarrow 1$ to 5 **do**
    $Tokenizer_i \leftarrow TrainTokenizer_i(\text{EN}_{\text{processed}}, \text{UR}_{\text{processed}})$
    $Representation_i \leftarrow Tokenizer_i(\text{EN-UR}_{\text{processed parallel}})$
    $Train_i, Val_i, Test_i \leftarrow SplitDataset(Representation_i)$
    $Model_i \leftarrow TrainDPETransformer(Train_i, Val_i)$
    $BLEU_i \leftarrow EvaluateTranslation(Model_i, Test_i)$
end for
$Model_{best} \leftarrow \text{argmax}(BLEU_i)$
Define candidate scaling parameters: $\lambda = [\lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5]$
$\lambda_{opt} \leftarrow \lambda_1$, $S_{opt} \leftarrow BLEU(Model_{best}, \lambda_1)$
for $j \leftarrow 2$ to 5 **do**
    Set scaling parameter of $Model_{best}$ to λ_j
    Retrain or fine-tune $Model_{best}$
$S_j \leftarrow EvaluateTranslation(Model_{best}, Test)$
    if $S_j > S_{opt}$ **then**
        $\lambda_{opt} \leftarrow \lambda_j$, $S_{opt} \leftarrow S_j$
    end if
end for
$ParameterOptimizedModel \leftarrow Model_{best}$ with λ_{opt}

English contractions were expanded (e.g., don't → do not), and all English characters were converted to lowercase. Urdu numerals were also standardized to handle multiple numeral forms consistently. Finally, Moses tokenization (Koehn et al., 2007) was applied to segment the corpus into tokens using spaces and punctuation as boundaries. Table 1 provides a summary of the dataset before and after preprocessing.

Table 1 Corpus Collection for Tokenizers Training

	English Words	Urdu Words
Unfiltered	621,816,259	334,542,952
Filtered	621,795,309	334,431,969

3.2 Parallel Dataset Collection and Preprocessing for Neural Machine Translation

To support the training of high-quality English–Urdu Neural Machine Translation models, a large-scale parallel corpus was compiled from diverse publicly available sources. The selected datasets span a wide range of domains including religious texts, news, subtitles, political speeches, legal documents, human rights, and general media content. This broad domain coverage ensures lexical richness and structural diversity, which are essential for building robust translation models. A detailed breakdown of each dataset source is provided below, with sentence-level statistics summarized in Table 2.

- **UMCOO5:** Parallel Bible and Quran corpus with sentence-level alignment in English and Urdu (Jawaid and Zeman, 2011).
- **CVIT PIB [v1.3]:** Sentence-aligned corpus from the Press Information Bureau website covering 11 Indian languages (Siripragada et al., 2020).
- **CVIT MKB [v0.0]:** Translations of Prime Minister's "Mann Ki Baat" radio speeches (Siripragada et al., 2020).
- **PIB Archives and The Wire:** Parallel corpora from Anuvaad platform across multiple domains such as news, legal, and healthcare (Ramesh et al., 2022).
- **English-Urdu:** Domain-specific corpora (cinema, news, subtitles, Bible) collected for Indian languages (Choudhary et al., 2020).
- **CLE:** Parallel corpus developed by the Center for Language Engineering (CLE), based on diverse newspaper content and manually translated into Urdu.[3]
- **UDHR:** Universal Declaration of Human Rights corpus in Unicode, sourced from OHCHR.[4]

[3] https://www.cle.org.pk/software/ling_resources/testingcorpusmt.htm

[4] https://unicode.org/udhr/index.html

Table 2 English-Urdu Parallel Corpus Collection for NMT

English-Urdu Dataset	Parallel Sentences
UMCOO5	Quran = 6,414, Bible = 7,957
CVIT PIB [v1.3]	202,578
CVIT MKB [v0.0]	1,018
PIB (2017-2020)	141,244
PIB Archives (2015-2016)	9,939
The wire (2021)	10,820
English-Urdu	40,453
CLE	400
UDHR	200
CC Matrix v1	6,100,000
Tatoeba	1,662
TED	15,754
Tico-19	3,071
QED	19,053
Global Voices	3,900
Total	**6,564,463**

- **OPUS:** Publicly available datasets from the OPUS repository, including CC Matrix v1, Tatoeba, TED, TICO-19, QED, and Global Voices (Tiedemann, 2012).

All collected corpora were subjected to a preprocessing pipeline to ensure the highest quality for training translation models. Initial steps such as symbolic content filtering, trimming and condensing of white-space, empty line removal, English contraction expansion, English case normalization, Urdu numeral normalization, and Moses tokenization were performed following the procedures detailed in Section 3.1. In addition to these, further filtration steps specific to the parallel dataset were applied.

To address inconsistencies in numerical data, a non-zero numeral filter was implemented. This filter ensures consistent translation formats for numerals used in quantities and dates by penalizing sentence pairs exhibiting inconsistent or disordered numerals. Numeral sequences were identified while preserving their positional information, accounting for cases such as omitted leading zeros. The final alignment score was computed using Python's difflib library, with a default minimum threshold of 0.5 for accepting valid sentence pairs.

Furthermore, filters based on the techniques proposed by Muischnek and Müürisep (2019) were utilized. The equal source-target filter eliminates sentences that are identical on both the source

Table 3 Statistics of English-Urdu Parallel Corpus

	English	Urdu
Parallel Sentences (Unfiltered)	6,564,463	6,564,463
Parallel Sentences (Filtered)	5,775,799	5,775,799
Words (Unfiltered)	78,676,618	98,482,406
Words (Filtered)	67,576,459	86,324,229

and target sides. The unique parallel sentence filter ensures that duplicate sentence pairs are removed, maintaining uniqueness across the corpus. The correct language filter discards sentences not belonging to the specified source or target languages, thereby preserving language consistency within the dataset.

For alignment quality, a multi-target alignment filter was developed to remove instances where a single source sentence was aligned to multiple target sentences. Similarly, a multi-source alignment filter was created to eliminate cases where a single target sentence corresponded to multiple source sentences.

Finally, filtering mechanisms from the Moses toolkit (Koehn et al., 2007) were incorporated to refine the corpus further. The word limit threshold filter discards sentence pairs where the word count is either less than one or exceeds one hundred words. The character limit threshold removes segments containing words with a character length exceeding fifty. Additionally, the sentence ratio limit filter eliminates sentence pairs that violate a maximum sentence length ratio of 1.7 between the source and target sides.

The filtered dataset is publicly available on Mendeley data (Asghar, 2025). The parallel number of parallel sentences and number of words before and after cleaning are mentioned in Table 3. The final cleaned corpus comprises 5,775,799 parallel sentence pairs, to the best of our knowledge, making it the largest, most refined, and publicly available English–Urdu dataset to date. Figure 1, illustrates the sentence length distributions of the English and Urdu corpora after preprocessing. Both languages exhibit highly similar, right-skewed distributions, with the majority of sentences concentrated between approximately 5 and 20 words. The close overlap between the distributions indicates comparable sentence-length characteristics across the two corpora.

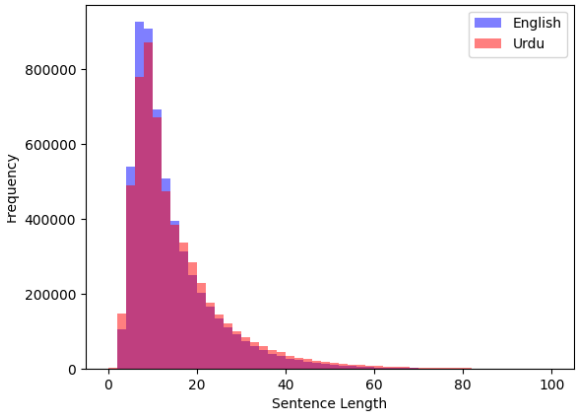


Fig. 1 English-Urdu Sentence Length Distribution

3.3 Subword Tokenization

To address vocabulary limitations in English–Urdu neural machine translation, five subword tokenization algorithms were evaluated. The Subword-nmt[5] and SentencePiece (Kudo and Richardson, 2018) libraries were selected based on their proven effectiveness and widespread adoption in prior NMT research. Specifically, Byte Pair Encoding (BPE) was trained using the `Subword-nmt` library, while the `SentencePiece` toolkit was used to train multiple models including BPE, word-based, unigram language model, and character-based tokenizers.

Separate vocabularies for English and Urdu were trained using each algorithm. To ensure a fair comparison, all tokenizers were trained from scratch on the preprocessed corpora with a fixed vocabulary size of 30,000 tokens and a character coverage of 1.0.

Once the tokenizers were trained, the filtered parallel corpus was segmented using the corresponding English–Urdu tokenizers. The outputs generated by the five selected tokenizers are illustrated in Figure 2 and Figure 3 and are discussed in Section 4.1.

3.4 Neural Machine Translation

After segmentation using the selected tokenizers, the resulting parallel corpora were divided into training, validation, and test sets using an 80%–10%–10% split, as shown in Table 4. These

[5] https://github.com/rsennrich/subword-nmt

English String:
`Hello, how are you ?`

Subword-nmt-BPE:
`['H@@', 'ell@@', 'lo', ',', 'how', 'are', 'you', '?']`

SentencePiece-BPE:
`['H', 'ello', '_,', '_how', '_are', '_you', '_?']`

SentencePiece-word:
`['Hell', 'o', '_,', '_ho', 'w', '_are', '_you', '_?']`

SentencePiece-unigram:
`['Hell', 'o', '_,', '_h', 'ow', '_are', '_you', '_?']`

SentencePiece-char:
`['H', 'e', 'l', 'l', 'o', '_,', '_h', 'o', 'w', '_a', 'r', 'e', '_y', 'o', 'u', '_?']`

Fig. 2 English sentence segmentation across five subword strategies

Urdu Sentence:
ہیلو، آپ کیسے ہیں؟

Subword-nmt-BPE:
['ہی@@', 'لو', '،', 'آپ', 'کیسے', 'ہیں', '؟']

SentencePiece-BPE:
['ہی', 'لو', '_،', '_آپ', '_کیس', 'ے', '_ہیں', '_؟']

SentencePiece-word:
['ہی', 'لو', '_،', '_آپ', '_کی', 'سے', '_ہیں', '_؟']

SentencePiece-unigram:
['ہ', 'ی', 'لو', '_،', '_آپ', '_کی', 'سے', '_ہیں', '_؟']

SentencePiece-char:
['ہ', 'ی', 'ل', 'و', '_،', '_آ', 'پ', '_ک', 'ی', 'س', 'ے', '_ہ', 'ی', 'ں', '_؟']

Fig. 3 Urdu sentence segmentation across five subword strategies

Table 4 Dataset Splitting

Split	Sentences
Training Set	4,620,639
Validation Set	577,579
Test Set	577,581

datasets served as the basis for training and evaluating the translation models.

3.4.1 Word Alignment

To capture word-order correspondences between English and Urdu, word alignments were generated using FastAlign (Dyer et al., 2013). These alignments establish token-level correspondences between the source and target sentences, providing the alignment information required by the adopted reordering-aware transformer.

For English-Urdu translation, each tokenized training set was processed with FastAlign using English as the source language and Urdu as the target language. For Urdu-English translation, the source and target languages were reversed. Since FastAlign requires access to both source and target sentences, word alignments were generated only for the training data. The resulting alignments are subsequently incorporated into the DPE framework, as described in Section 3.4.2.

3.4.2 Transformer Model with Dynamic Position Encoding

To address syntactic divergence between English (SVO) and Urdu (SOV), this study adopted the dynamic position encoding based transformer architecture (Zheng et al., 2022), which refines source positional embeddings to reflect the target-language word order without explicitly reordering the input sequence. Unlike the fixed positional encodings used in the standard Transformer, the variant model learns to adapt source positional information during training through an auxiliary reordering objective guided by word alignment supervision. This enables the model to capture the word-order transformations required between English and Urdu while preserving the original source sequence.

Each source word embedding w_i is first combined with its sinusoidal positional encoding p_i (Vaswani et al., 2017). The resulting representation, $w_i \oplus p_i$, is passed through a two-layer DPE Transformer network that generates a refined positional embedding r_i. These refined positional embeddings replace the original positional encodings before being supplied to the first encoder layer of the translation model. In contrast to the standard Transformer, which uses fixed sinusoidal positional encodings throughout encoding, this position aware model replaces them with refined positional embeddings generated by its network.

During training, this network is supervised using an auxiliary reordering objective. The word alignments generated by FastAlign (Dyer et al., 2013) provide token-level correspondences between the source and target sentences. For each source word w_i, the aligned target word determines the reference target position index j, from which the corresponding sinusoidal positional embedding p_j is obtained. The reordering loss is computed as the Mean Squared Error (MSE) between the refined positional embedding

r_i and the target positional embedding p_j:

$$\mathcal{L}_{\text{reorder}} = \frac{1}{|S|} \sum_{i=1}^{|S|} \text{MSE}(p_j, r_i) \quad (1)$$

where $|S|$ denotes the source sequence length. The overall training objective combines the translation loss and the auxiliary reordering loss:

$$\mathcal{L}_{\text{total}} = \lambda \mathcal{L}_{\text{translation}} + (1 - \lambda)\mathcal{L}_{\text{reorder}} \quad (2)$$

The hyperparameter λ controls the balance between translation quality and positional refinement. Lower values increase the relative contribution of the reordering objective, whereas higher values place greater emphasis on the translation objective.

Since FastAlign requires simultaneous access to both the source and target sentences to generate word alignments, it is used only during training to supervise the auxiliary reordering objective. Once training is complete, the network predicts refined positional embeddings directly from the source sentence, eliminating the need for target references or external alignment tools during inference. Consequently, the model performs translation in an end-to-end manner while preserving the word-order knowledge learned during training.

For each of the five tokenization variants described in Section 3.3, a separate adopted transformer model was trained. The best-performing tokenizer-model combination for each translation direction was subsequently selected for hyperparameter tuning of λ, as described in Section 3.4.3. Translation quality was evaluated using BLEU (Bilingual Evaluation Understudy) (Papineni et al., 2002), which remains the most widely used metric in English–Urdu machine translation research and facilitates direct comparison with previously published studies. For Urdu–English translation, additional evaluation metrics including METEOR (Metric for Evaluation of Translation with Explicit Ordering) (Banerjee and Lavie, 2005), TER (Translation Edit Rate) (Snover et al., 2006), and NIST (National Institute of Standards and Technology) (Doddington, 2002) were also reported to provide a broader assessment of translation quality.

3.4.3 Reordering Loss Weight Optimization

Following the initial training of Transformer models, further fine-tuning was conducted by adjusting the reordering loss weight λ in Equation 2, which controls the contribution of the auxiliary reordering loss relative to the main translation loss during optimization.

Multiple values of λ were evaluated, ranging from 0.1 to 0.7 in increments of 0.2, following the experimental setup of Zheng et al. (2022). This evaluation was conducted to determine the most effective balance between translation learning and reordering supervision for the English–Urdu translation task. To verify its suitability for the English–Urdu translation task, preliminary validation experiments were also conducted using values outside this interval. These experiments consistently showed degraded performance, with $\lambda < 0.1$ reduced translation quality and $\lambda > 0.7$ weakened positional refinement. Consequently, only values within the range of 0.1 to 0.7 were considered for the final hyperparameter search. The selected range therefore reflects both prior work and empirical validation specific to our task.

The value that minimized the total loss while maximizing translation accuracy was selected for final evaluation. This tuning process ensured a balanced contribution of both translation quality and alignment learning, especially important for language pairs with differing word order structures like English and Urdu.

3.4.4 Experimental Setup

All experiments were implemented using the Fairseq toolkit (Ott et al., 2019). The baseline employed the Transformer Base architecture, while the main experiments utilized the Transformer variant (Zheng et al., 2022). Training was performed for 20 epochs using the Adam optimizer with an inverse square-root learning-rate schedule on an NVIDIA Tesla T4 GPU. Unless otherwise specified, default Fairseq Transformer Base hyperparameters were used. The best checkpoint was selected based on validation BLEU score. During inference, beam search with a beam size of 5 and length penalty of 1.0 was applied. All experiments used a fixed random seed of 42.

4 Results and Discussion

This section presents the experimental results and their analysis for each stage of the proposed approach, following the data preprocessing and word alignment steps. It begins with a comparison of different subword tokenization methods to address vocabulary sparsity, while analyzing the adopted transformer model performance trained on each tokenized dataset. The section then examines the effect of hyperparameter tuning, specifically the optimization of the scaling parameter. The results highlight the impact of tokenization choices and positional encoding optimization on English and Urdu translation quality.

4.1 Tokenizer Evaluation and Translation Results

Five tokenization strategies were evaluated for English-Urdu and Urdu-English neural machine translation using a fixed model configuration. All models were trained on the same preprocessed parallel corpus to ensure that performance differences were attributable solely to tokenization choices.

As shown in Table 5, Subword-NMT BPE achieved the highest BLEU score (30.30) for English-Urdu translation, while SentencePiece-BPE produced the best BLEU score (26.90) for Urdu-English (Table 6). Across both translation directions, BPE-based approaches consistently outperformed word-level, unigram, and character-level tokenization methods. The observed differences can be partially explained by the segmentation patterns in Figures 2 and 3. For English-Urdu translation, the superior performance of Subword-NMT BPE can be attributed to its ability to preserve semantically meaningful subword units while decomposing infrequent words into reusable components.

Figure 2 illustrates that words are segmented into meaningful fragments rather than isolated characters, allowing the model to represent rare and frequent words through reusable subword units while reducing vocabulary sparsity. This is particularly beneficial during Urdu generation, where preserving semantic information across subword boundaries may contribute to more accurate target-language generation. In contrast, SentencePiece-BPE produced the best results for Urdu-English translation. Figure 3 shows that SentencePiece-BPE learns segmentation directly from the training corpus while explicitly encoding word-boundary information, providing richer source-side representations of Urdu text and enabling the encoder to capture contextual dependencies more effectively before translation into English. These findings suggest that English-Urdu translation benefits more from preserving coherent subword units during target-language generation, whereas Urdu-English translation benefits more from robust source-side encoding through corpus-driven segmentation. Word-level tokenization remained vulnerable to unseen vocabulary because rare words could not be decomposed into reusable units, while character-level tokenization substantially increased sequence length by fragmenting meaningful units across many tokens, making contextual modeling more difficult.

Collectively, these findings highlight the effectiveness of BPE-based tokenization in reducing vocabulary sparsity through meaningful subword segmentation. Although this comparison focused on tokenization strategies, translation quality

Table 5 Translation performance for English-Urdu across different tokenization techniques

Tokenization Method	BLEU
Subword-nmt-BPE	**30.30**
SentencePiece-BPE	30.11
SentencePiece-Word	29.86
SentencePiece-Unigram	29.01
SentencePiece-Character	28.83

Table 6 Translation performance for Urdu-English across different tokenization techniques

Tokenization Method	BLEU	METEOR	TER	NIST
Subword-nmt-BPE	26.08	0.463	0.535	0.911
SentencePiece-BPE	**26.90**	0.459	0.534	0.912
SentencePiece-Word	26.47	0.456	0.549	0.901
SentencePiece-Unigram	24.45	0.421	0.510	0.934
SentencePiece-Character	25.01	0.376	0.547	0.900

Table 7 Impact of the reordering loss weight λ on English-Urdu translation performance

Model	BLEU
Transformer Base Vaswani et al. (2017)	28.78
Transformer-DPE ($\lambda = 0.1$)	29.01
Transformer-DPE ($\lambda = 0.3$)	30.30
Transformer-DPE ($\lambda = 0.5$)	**31.95**
Transformer-DPE ($\lambda = 0.7$)	29.91

Table 8 Impact of the reordering loss weight λ on Urdu-English translation performance

Model	BLEU
Transformer Base Vaswani et al. (2017)	25.50
Transformer-DPE ($\lambda = 0.1$)	26.00
Transformer-DPE ($\lambda = 0.3$)	26.90
Transformer-DPE ($\lambda = 0.5$)	27.70
Transformer-DPE ($\lambda = 0.7$)	**27.90**

was also influenced by the adopted Transformer architecture, which addresses word-order divergence between English and Urdu. Even when using the same tokenizer, the transformer variant outperformed the standard transformer baseline by improving BLEU scores, as discussed in Section 4.2.

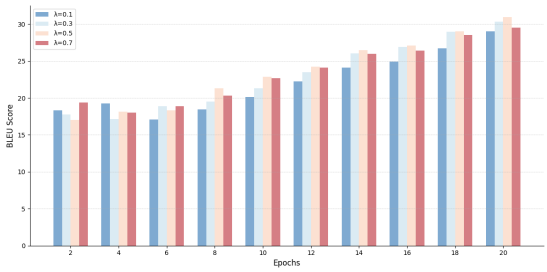
4.2 Hyperparameter Tuning Results

To further enhance model performance, particularly in capturing reordering patterns between English and Urdu, hyperparameter tuning was conducted by adjusting the positional encoding loss weight λ in the model. The objective was to determine the optimal balance between translation accuracy and position prediction, as governed by the total loss function defined in Equation 2. This experiment was conducted by fine-tuning the best-performing trained models obtained from the tokenizer evaluation stage, corresponding to *Subword-nmt-BPE* for English–Urdu and *SentencePiece-BPE* for Urdu–English, using a default λ value of 0.3. These choices ensured that hyperparameter optimization was applied to the most effective tokenization configurations for each translation direction.

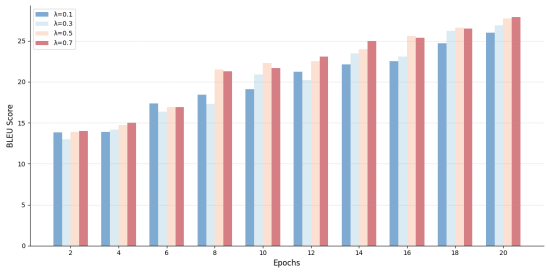
Tables 7 and 8 report BLEU scores across different λ values, alongside the standard Transformer baseline. For English-Urdu translation, a λ value of 0.5 yielded the highest BLEU score of 31.95. Conversely, for Urdu-English, the best performance was achieved at $\lambda = 0.7$, with a BLEU score of 27.90.

These results demonstrate the importance of modeling positional dependencies and fine-tuning positional encoding weights. Appropriate selection of λ significantly improves translation quality by enabling more effective modeling of syntactic reordering patterns inherent in English–Urdu language pairs.

In addition to the final BLEU scores, Figure 4 provides a visual trajectory of model performance across training epochs for each λ setting. These trends reveal that while all models improved steadily over time, certain configurations demonstrated more accelerated gains. For English-Urdu translation, $\lambda = 0.5$ consistently led to higher BLEU scores beyond epoch 8, confirming its superior balance between translation and position prediction losses. In the Urdu-English direction, $\lambda = 0.7$ maintained a clear advantage in later epochs, reflecting its effectiveness in capturing complex reordering patterns.



(a) BLEU score vs. epochs for EN-UR



(b) BLEU score vs. epochs for UR-EN

Fig. 4 BLEU score comparison across epochs for different λ values

Table 9 Performance comparison between the baseline and the proposed approach

Approach	EN–UR	UR–EN	Sent. (k)
Transformer Base Vaswani et al. (2017)	28.78	25.50	5,776
Transformer-DPE ($\lambda = 0.5$)	**31.95**	**27.90**	**5,776**

Table 10 Existing English and Urdu Machine Translation Approaches

Approach	EN–UR	UR–EN	Sent. (k)
SMT Ali et al. (2013)	37.10	–	27
LSTM Andrabi et al. (2022)	45.83	–	31
MNMT Sen et al. (2018)	21.48	26.26	27
CEUT Rai et al. (2020)	29.94	–	648
MNMT Bala Das et al. (2023)	28.70	26.90	5,520
LKMT Ul Hassan et al. (2024)	30.90	29.86	648
FConv-NN Israr et al. (2024)	–	40.22	10
Transformer (Dual-Encoder) Israr et al. (2025)	–	43.10	100
Transformer (Avg. Embeddings) Zuhra et al. (2026)	47.18	–	1083

4.3 Performance Analysis

Table 9 presents the performance of our optimized Transformer configuration compared with the Transformer baseline on the identical dataset. The optimized configuration achieves BLEU scores of 31.95 for English-Urdu and 27.90 for Urdu-English, outperforming the baseline by +3.17 and +2.40 BLEU points, respectively. These improvements indicate that positional encoding optimization contributes positively to translation quality.

To further examine the impact of positional encoding optimization on word-order divergence, Figure 5 presents representative English-Urdu and Figure 6 presents representative Urdu-English translation examples. The highlighted segments indicate differences in constituent ordering between the baseline and the optimized configuration. While the baseline often preserves source language ordering patterns, the optimized configuration generally produces structures that better align with the target language syntax. In the English-Urdu examples (Figure 5), the baseline places temporal phrases such as "after several rounds of discussion" and "next month" at the sentence-final position, preserving English post verbal adjunct ordering. The optimized configuration correctly fronts these temporal constituents to the pre verbal slot, yielding fluent Urdu SOV structure. In the Urdu-English examples (Figure 6), the baseline retains Urdu-like ordering, placing the temporal clause "after evaluating the evidence" immediately after the subject and positioning the prepositional phrase "with several countries" before the infinitive "to collaborate". The optimized model corrects these errors, producing natural English SVO order with proper

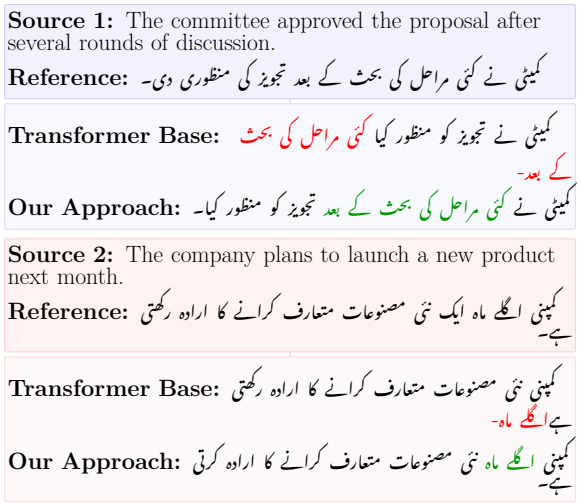


Fig. 5 Qualitative comparison of English–Urdu translation outputs

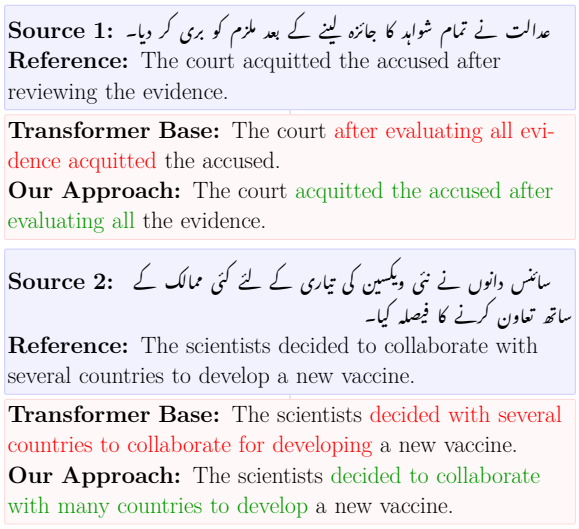


Fig. 6 Qualitative comparison of Urdu-English translation outputs

adjunct placement. Although occasional lexical and grammatical variations remain, the examples demonstrate improved handling of cross lingual reordering, consistent with BLEU gains observed in Table 9.

Table 10 provides a contextual comparison with previously reported English–Urdu machine translation systems. Direct comparison of BLEU scores across studies should be interpreted cautiously because translation performance is influenced by differences in dataset composition, domain coverage, preprocessing pipelines, and evaluation settings (Post, 2018). Furthermore, the studies summarized in Table 10 were trained and evaluated on datasets that vary substantially in size, ranging from 10k to 5.52 million sentence pairs. Previous work has shown that translation quality is highly sensitive to the quantity and characteristics of training data, making cross-study BLEU comparisons inherently challenging (Miceli Barone et al., 2017; Lankford et al., 2021). Nevertheless, among approaches trained on large-scale datasets, the proposed framework achieves competitive performance while utilizing a publicly released corpus comprising 5.77 million parallel sentence pairs. These results highlight the complementary roles of effective subword tokenization and reordering-aware word-order modeling in improving English–Urdu neural machine translation.

5 Threats to Validity

Translation quality was evaluated using established automatic metrics, including BLEU, METEOR, TER, and NIST. While these metrics facilitate objective comparison with prior studies, they may not fully reflect all aspects of translation quality as perceived by human evaluators. Future work may incorporate language-agnostic metrics such as chrF, statistical validation techniques including significance testing and confidence intervals, and human assessment to provide a more comprehensive evaluation for English–Urdu translation systems.

6 Conclusion

This study presented a framework for improving English–Urdu neural machine translation by addressing two key challenges: vocabulary sparsity and syntactic word-order divergence. To support reproducible research, a large-scale English–Urdu parallel corpus was compiled, cleaned, and released publicly. The proposed framework experimentally evaluated multiple subword tokenization strategies and a reordering-aware Transformer architecture to mitigate vocabulary limitations and structural differences between English and Urdu.

Experimental results demonstrated that tokenization strategy plays a critical role in translation quality. Subword-NMT BPE achieved the best performance for English–Urdu translation, while SentencePiece-BPE produced the strongest results for Urdu–English translation. In addition, optimizing the reordering loss weight consistently improved performance over the Transformer baseline in both translation directions. These findings highlight the importance of addressing vocabulary segmentation and word-order divergence, demonstrating that targeted adaptations of Transformer-based models can yield meaningful improvements for linguistically distant language pairs.

From a practical perspective, the proposed framework has potential applications in multilingual communication, content localization, educational technologies, and cross-lingual information access for Urdu-speaking communities. Practical deployment may additionally require addressing challenges related to domain adaptation, computational efficiency, and robustness to noisy user-generated text.

7 Declarations

Conflicts of interest: The authors declare no conflict of interest. This study was not funded by any agency or organization.

References

Ali A, Hussain A, Malik MK (2013) Model for english-urdu statistical machine translation. World Applied Sciences 24:1362–1367

Andrabi SAB, Wahid A, et al (2022) Machine translation system using deep learning for english to urdu. Computational intelligence and neuroscience 2022

Asghar R (2025) EN-UR Parallel Dataset. Mendeley Data, V2, https://doi.org/10.17632/

bmh9xjyjgb.2
Bala Das S, Biradar A, Kumar Mishra T, et al (2023) Improving multilingual neural machine translation system for indic languages. ACM Transactions on Asian and Low-Resource Language Information Processing 22(6):1–24
Banerjee S, Lavie A (2005) Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In: Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization, pp 65–72
Choudhary H, Rao S, Rohilla R (2020) Neural machine translation for low-resourced Indian languages. In: Proceedings of the Twelfth Language Resources and Evaluation Conference. European Language Resources Association, Marseille, France
Craciunescu O, Gerding-Salas C, Stringer-O'Keeffe S (2004) Machine translation and computer-assisted translation. Machine Translation and Computer-Assisted Translation
Doddington G (2002) Automatic evaluation of machine translation quality using n-gram co-occurrence statistics. In: Proceedings of the second international conference on Human Language Technology Research, pp 138–145
Dyer C, Chahuneau V, Smith NA (2013) A simple, fast, and effective reparameterization of ibm model 2. In: Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp 644–648
Eberhard DM, Simons GF, Fennig CD (2026) Urdu. URL https://www.ethnologue.com/language/urd, accessed 11 July 2026
Israr H, Shahzad MK, Anwar S (2024) Improved urdu-english neural machine translation with a fully convolutional neural network encoder. International Journal of Mathematical, Engineering & Management Sciences 9(5)
Israr H, Khan SA, Tahir MA, et al (2025) Knowledge-grounded attention-based neural machine translation model. Applied Computational Intelligence and Soft Computing 2025(1):6234949
Jawaid B, Zeman D (2011) Word-order issues in english-to-urdu statistical machine translation. Prague Bull Math Linguistics 95:87–106
Jindal S, Goyal V, Bhullar JS (2018) English to punjabi statistical machine translation using moses (corpus based). Journal of Statistics and Management Systems 21(4):553–560
Khan S, Mir U, Shreem SS, et al (2018) Translation divergence patterns handling in english to urdu machine translation. International Journal on Artificial Intelligence Tools 27(05):1850017
Ko WJ, El-Kishky A, Renduchintala A, et al (2021) Adapting high-resource NMT models to translate low-resource related languages without parallel data. In: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pp 802–812
Koehn P, Hoang H, Birch A, et al (2007) Moses: Open source toolkit for statistical machine translation. In: Proceedings of the 45th annual meeting of the association for computational linguistics companion volume proceedings of the demo and poster sessions, pp 177–180
Kudo T, Richardson J (2018) Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations. Association for Computational Linguistics, Brussels, Belgium, pp 66–71
Lankford S, Afli H, Way A (2021) Machine translation in the covid domain: an EN↔GA case study for LoResMT 2021. In: Proceedings of the 4th Workshop on Technologies for MT of Low Resource Languages (LoResMT2021), pp 144–150
Luo Y, Xu Z, Ouyang Y, et al (2026) Niutrans. lmt: Toward inclusive and scalable multilingual machine translation with llms. In: Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp 25151–25179
Luong M, Sutskever I, Le QV, et al (2015) Addressing the rare word problem in neural machine translation
Mahesh R, Sinha K (2009) Developing english-urdu machine translation via hindi. In: Proceedings of the Third Workshop on Computational Approaches to Arabic-Script-based Languages (CAASL3)

Miceli Barone AV, Haddow B, Germann U, et al (2017) Regularization techniques for fine-tuning in neural machine translation. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing

Muischnek K, Müürisep K (2019) Impact of corpora quality on neural machine translation. In: Human Language Technologies–The Baltic Perspective: Proc. of the Eighth International Conference Baltic HLT 2018, IOS Press, p 126

Ott M, Edunov S, Baevski A, et al (2019) fairseq: A fast, extensible toolkit for sequence modeling. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations). Association for Computational Linguistics, Minneapolis, Minnesota, pp 48–53

Papineni K, Roukos S, Ward T, et al (2002) Bleu: a method for automatic evaluation of machine translation. In: Proceedings of the 40th annual meeting of the Association for Computational Linguistics, pp 311–318

Post M (2018) A call for clarity in reporting BLEU scores. In: Proceedings of the Third Conference on Machine Translation (WMT), pp 186–191

Rai SI, Khan MU, Anwar MW (2020) English to urdu: Optimizing sequence learning in neural machine translation. In: 2020 3rd international conference on computing, mathematics and engineering technologies (iCoMET), IEEE, pp 1–6

Ramesh G, Doddapaneni S, Bheemaraj A, et al (2022) Samanantar: The largest publicly available parallel corpora collection for 11 indic languages. Transactions of the Association for Computational Linguistics 10:145–162

Saini S, Sahula V (2020) Chapter 11 - setting up a neural machine translation system for english to indian languages. In: Sinha G, Suri JS (eds) Cognitive Informatics, Computer Modelling, and Cognitive Science. Academic Press, Cambridge, MA, p 195–212

Sen S, Gupta KK, Ekbal A, et al (2018) Iitp-mt at wat2018: Transformer-based multilingual indic-english neural machine translation system. In: Proceedings of the 32nd Pacific Asia conference on language, information and computation: 5th workshop on Asian translation: 5th workshop on Asian translation

Siripragada S, Philip J, Namboodiri VP, et al (2020) A multilingual parallel corpora collection effort for Indian languages. In: Proceedings of the Twelfth Language Resources and Evaluation Conference, pp 3743–3751

Snover M, Dorr B, Schwartz R, et al (2006) A study of translation edit rate with targeted human annotation. In: Proceedings of the 7th Conference of the Association for Machine Translation in the Americas: Technical Papers. Association for Machine Translation in the Americas, Cambridge, Massachusetts, USA, pp 223–231

Tiedemann J (2012) Parallel data, tools and interfaces in opus. In: Lrec, Citeseer, pp 2214–2218

Ul Hassan MN, Yu Z, Wang J, et al (2024) Lkmt: Linguistics knowledge-driven multi-task neural machine translation for urdu and english. Computers, Materials & Continua 81(1)

Vaswani A, Shazeer N, Parmar N, et al (2017) Attention is all you need. Advances in neural information processing systems 30

Visweswariah K, Rajkumar R, Gandhe A, et al (2011) A word reordering model for improved machine translation. In: Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, pp 486–496

Wagh B, Bapat H, Gupta N, et al (2025) Transformers: Leveraging OpenNMT and transfer learning for low-resource Indian language translation. In: Haddow B, Kocmi T, Koehn P, et al (eds) Proceedings of the Tenth Conference on Machine Translation. Association for Computational Linguistics, Suzhou, China, pp 1227–1232

Zheng J, Rezagholizadeh M, Passban P (2022) Dynamic position encoding for transformers. In: Calzolari N, Huang CR, Kim H, et al (eds) Proceedings of the 29th International Conference on Computational Linguistics. International Committee on Computational Linguistics, Gyeongju, Republic of Korea, pp 5076–5084

Zuhra FT, Din SJU, Ali H, et al (2026) English to urdu neural machine translation using transformer with averaged word embeddings. VAWKUM Transactions on Computer Sciences 14(1):1–14