VAWKUM Transactions on Computer Sciences
http://vfast.org/journals/index.php/VTCS@ 2026, ISSN(e): 2308-8168, ISSN(p): 2411-6335
Volume 14, Number 1, January-June 2026 pp:01-14

English to Urdu Neural Machine Translation using Transformer with Averaged Word Embeddings

Fatima Tuz Zuhra[1*], **Syed Jamal Ud Din**[2], **Hina Ali**[3], **Surayya Naz**[4], **Sumaira Rasool**[5], **Fouzia Idrees**[6]

[1]Department of Computer Science, University of Peshawar, Pakistan; [2]Institute of Health Sciences, University of Peshawar, Pakistan; [3]Department of Computer Science, National University of Modern Languages, Islamabad Campus, Pakistan; [4]Department of Computer Science, Abdul Wali Khan University Mardan, Pabbi Campus, Pakistan; [5]Department of Computer Science, University of Peshawar, Pakistan; [6]Department of Computer Science, Shaheed Benazir Bhutto Women University Peshawar, Pakistan

Keywords:*Neural Machine Translation, Transformer, Word Embeddings, English–Urdu Translation, Low-Resource NLP, BLEU Evaluation*

Journal Info:
Submitted:
August 09, 2025
Accepted:
February 10, 2026
Published:
February 22, 2026

ABSTRACT

In this era of computers, the World Wide Web (WWW) content is vital for everyone. Most of the useful content on the web is in English. Machine translation from the web content into the national language of Pakistan, i.e. Urdu, has several applications such as Urdu text generation, language resources creation, language research and availability of knowledge to illiterate individuals who cannot comprehend English but know Urdu. We propose a novel English-to-Urdu machine translation model in this research work, based on the transformer model that exploits the average of three word embeddings. These three word embeddings are Urdu word2vec (skipgram-based), part-of-speech-ngram (POS-Ngram) embeddings, and POS-POS embeddings, both of which encode the rich morphological and morphosyntactic features of Urdu language inside the word embeddings. Experiments are performed using a manually compiled English-Urdu parallel corpus from OPUS corpora and Github. The proposed transformer-based approach is compared to fine-tuned Llama-3-8B, T5-small, Long Short-Term Memory (LSTM), and Bi-directional LSTM (Bi-LSTM). The evaluation metrics used are BLEU and ROUGE-L scores. The results suggest that the proposed model outperforms T5-small, LSTM and Bi-LSTM by ≈ 2.15, 7.44 and 5.53 points respectively, in BLEU score and by ≈ 1.7, 2.5 and 4.12 points, respectively, in ROUGE-L score. The proposed model shows comparable performance to the fine-tuned Llama-3-8B.

***Correspondence author email address:** fatima@uop.edu.pk
DOI: 10.21015/vtcs.v14i1.2213

1 Introduction

Urdu is the widely spoken language and national language of Pakistan. It is also one of the major spoken languages of India. It is one of the 22 major languages of India [1]. The language has over 231 million speakers [2], making it one of the most widely spoken languages in South Asia and among the top ten most spoken

languages in the world. Despite its large speaker base spanning multiple countries including Pakistan, India, and diaspora communities worldwide, Urdu remains significantly under-represented in digital content and computational linguistics research.

The majority of Urdu speakers in Pakistan are either illiterate or can poorly understand English language. According to educational statistics, a significant portion of Pakistan's population has limited exposure to English language instruction, and even among those who have received formal education, English proficiency remains concentrated in urban areas and among privileged segments of society. There is a need for English-to-Urdu machine translation in order to take full advantage of the knowledge available on the world wide web through the Internet. The World Wide Web is full of information about every subject such as education and healthcare to technology and entertainment, but the vast majority of the materials are not available to the Urdu speakers who do not know English. With correct machine translation systems, this digital divide can be bridged and the access to information democratized to millions of people.

Accurate machine translation needs language resources such as corpora, treebanks, POS taggers, and word embeddings [1, 3, 4]. These are resources that are the underlying infrastructure on which natural language processing systems are based. Parallel corpora contain aligned pairs of sentences in both source language and target languages in which computers learn how to translate. Syntactic annotations are provided in treebanks to provide models with information about grammatical structures. POS taggers determine grammatical categories of words whereas word embeddings are used to determine semantic relations between words in continuous vectors. Urdu is however a resource poor language where there are few and little good quality digital texts materials in comparison to other languages such as English, Chinese or Arabic. This is further complicated by the fact that although Urdu text is in existence; it most of the time occurs in the Romanized version where the Urdu language is written using the Roman script instead of the normal Perso-Arabic script. Such Romanization is widespread in online interactions, social networks and messaging systems where users find it easier to type using English keyboards. For example, the work of Ali and his co-authors [5] deals with age and gender identification inside romanized Urdu text, highlighting both the prevalence of Romanized Urdu and the need for NLP tools that can handle this phenomenon.

Machine translation is a field of natural language processing (NLP) that has witnessed remarkable progress over the past decade. The modern NLP techniques are based on the use of neural networks, which have revolutionized the field by enabling end-to-end learning of complex linguistic mappings. Unlike traditional statistical machine translation systems that relied on separate components for translation, language modeling, and reordering, neural machine translation systems learn to translate in a unified manner, capturing subtle linguistic nuances through distributed representations. A major milestone in the field of neural networks is achieved with the introduction of self-attention which enables neural networks to access distant contextual information more effectively than traditional recurrent architectures [6–9]. Self-attention enables models to consider the relevance of various words in a sentence irrespective of distance which is a weakness of recurrent neural networks that faced challenges with long-range dependencies because of vanishing gradient issues. This process calculates the scores of attention between one pair of position in a sequence to another, forming a direct pathway in the flow of information among the far words. This mechanism computes attention scores between all pairs of positions in a sequence, creating a direct pathway for information flow between distant words.

Transformers are the latest generation neural networks where the self-attention mechanism is used as a fundamental computational unit. Such models are constructed as sequence-to-sequence (seq2seq) networks that implies that they can change one sequence (as an input) to another sequence (as an output). The transformer architecture consists of an encoder that processes the input sequence and a decoder that generates the output sequence, with multiple layers of self-attention and feed-forward networks enabling rich representations at multiple levels of abstraction. This architectural paradigm is particularly well-suited for machine translation tasks, where sentences in a

source language must be mapped to their equivalent meanings in a target language. Figure 1 shows the use of seq2seq model for English to Urdu machine translation, demonstrating how input English text is processed and transformed into corresponding Urdu output.

Word embeddings are used to represent individual words in the form of vectors of real values [6] and are provided to machine learning models as input. They capture the context information of words in the text i.e. the accompanying words of a given word, based on the distributional hypothesis which posits that words appearing in similar contexts tend to have similar meanings. These vector representations enable mathematical operations on words, such that relationships like "king - man + woman = queen" can be approximated in the embedding space. Word embedding models such as word2vec [10] are trained based on words and its accompanying words, learning to predict a word from its context (continuous bag-of-words model) or predict the context from a target word (skip-gram model). The training process involves adjusting vector representations to minimize prediction error on large text corpora, resulting in embeddings that capture both semantic and syntactic relationships. Chen and Manning [11] proposed the idea of training word embeddings on other information such as POS in addition to words, and hence capture more linguistic information within the embedding space. Their work demonstrated that incorporating syntactic information during embedding learning leads to improved performance on downstream tasks like dependency parsing, as the embeddings encode not only word meanings but also their grammatical functions and contextual behavior.

2 State of the Art

NLP community in recent years have seen increasing interest in English-Urdu MT, but the field remains low-resource, with many challenges due to data scarcity, morphological richness, and orthographic complexity of Urdu language. Urdu morphology is particularly challenging for machine translation due to its rich system of inflectional and derivational morphology, including gender, number, case, and verb agreement markers that must be correctly generated in the target language. The script itself, a form of Nastaliq script, where it is written in the right-left direction with a complicated form of ligature, poses additional challenges of preprocessing and tokenization that do not exist in languages with Latin scripts. These problems are pointed out in a recent survey by Basit [12] and it is noted that most of the modern MT systems on Urdu language are hampered with absence of large high-quality of parallel corpora and sufficient evaluation. Some of the usual patterns of errors and difficulties peculiar to Urdu revealed by the survey are the processing of case markers, verb morphology, complex structures, and the widespread problem of the out-of-vocabulary words because of the rich morphology of the language.

The current state-of-the-art research within the sphere relies on encoder-decoder models using attention mechanism. The analysis carried out by Israr and his collaborators [13] demonstrates that learned evaluation measures can be effective and more efficient (in decoding) in low-resource MT, and that convolutional encoders can be competitive. Their work shows that through a fully convolutional architecture, it is possible to make large improvements in BLEU over a CNN baseline as well as exhibit better ability to capture long-range dependencies. Hassan and his co-authors [4] emphasize that the linguistically informed models can alleviate the problem of data scarcity by injecting the prior knowledge into the learning procedure. Their contribution to the knowledge of linguistics based multi-task neural machine translation demonstrates that learning to make predictions of POS tags, morphological features and translations in a multi-task structure results in better extrapolation of the results of the data with a limited amount. Nonetheless, model complexity and training has a trade-offs since, the higher the linguistic knowledge is integrated into the model, the more the model complexity and thus, the slower the training process, making it necessary to balance computational resources against the performance benefits. The language pairs can also be translated by the machine translation which can be done unsupervised and uses the monolingual corpora and approaches like back-translation to generate synthetic parallel data [2]. While unsupervised methods typically achieve lower BLEU scores compared

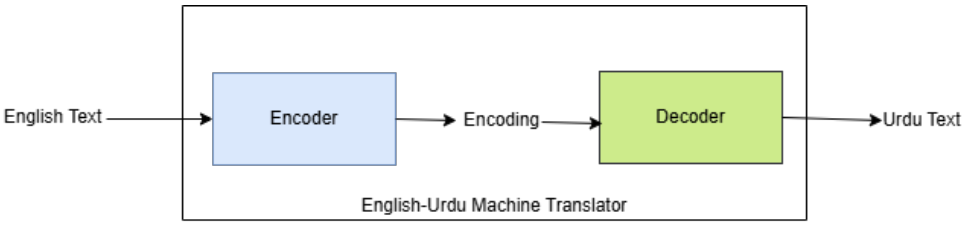


Figure 1. English to Urdu machine translation using seq2seq model

to supervised approaches, they offer a viable path forward for language pairs where parallel data is extremely scarce or non-existent.

An overview of the latest notable work in the field of English-Urdu machine translation is provided in Table 1, which summarizes key approaches, model architectures, and findings from recent studies. The table organizes work chronologically and highlights the evolution of techniques from traditional LSTM-based approaches to modern transformer architectures and large language models, while also documenting the persistent challenges that remain in English-Urdu machine translation.

We propose a transformer-based English-to-Urdu translator which exploits the average of three types of word embeddings. These three word embeddings are Urdu word2vec (skip-gram-based), part-of-speech-ngram (POS-Ngram) embeddings, and POS-POS embeddings, both of which encode the rich morphological and morphosyntactic features of Urdu language inside the word embeddings. The first embedding type captures general semantic relationships based on word co-occurrence patterns. The second type, POS-Ngram embeddings, combines part-of-speech information with character n-grams to capture morphological structure, which is crucial for Urdu's rich inflectional system. The third type, POS-POS embeddings, focuses on syntactic relationships by modeling sequences of POS tags, capturing grammatical patterns that are essential for producing syntactically correct Urdu output. By averaging these three complementary embedding types, we create a unified representation that leverages semantic, morphological, and syntactic information simultaneously. To the best of our knowledge, this is the first time that a transformer with averaged word embeddings, two of which capture the rich morphology and syntax of Urdu language, is used for English-to-Urdu machine translation. This innovative method builds on the advantages of several embedding methods to establish stronger and linguistically sensitive representations of the Urdu text, and this could help to counter certain constraints of data scarcity.

The remaining part of this paper has the following structure. Section 2 describes the proposed methodology in greater detail, such as the compilation of datasets, preprocessing steps, the generation of three types of embeddings, model architecture specification and hyperparameters. Section 3 contains the description of the obtained results in case the experiments are conducted, with comparison of the results with the baseline models and the analysis of BLEU and ROUGE-L scores. Part 4 ends the article with a conclusion of findings and the further research directions.

3 The Proposed Methodology

We propose the use of transformer-based model for English-to-Urdu machine translation, which are the state of the art in machine learning, and are based on attention mechanism. Transformers have revolutionized the field of natural language processing by enabling parallel processing of sequences and capturing long-range dependencies through self-attention, making them particularly suitable for machine translation tasks where understanding context across entire sentences is crucial for producing accurate translations. In this section we explore the methodology used in this research work, including dataset compilation and preprocessing, the architecture of the proposed model, the three types of word embeddings employed, tokenization approaches, hyperparameter settings, and experimental setup for evaluation.

Table 1. Recent English–Urdu Machine Translation Research

Work	Task / Direction	Model / Approach	Key Findings
[12]	EN → UR	Evaluation of GPT-3.5, opus-mt-en-ur, NLLB, IndicTrans2	IndicTrans2 performs best; highlights MT challenges for Urdu and common error patterns.
[3]	UR → EN	Fully convolutional encoder (FConv-NN)	Significant BLEU improvement over CNN baseline; better modeling of long dependencies.
[14]	EN ↔ UR	Linguistically informed multi-task NMT	Morphology/syntax tasks improve NMT quality for low-resource Urdu.
[15]	UR → EN	Unsupervised MT with back-translation	Unsupervised BLEU is low; supervised variant improves but still limited due to data scarcity.
[16]	UR → EN	Qualitative analysis of Google Translate evolution	Persistent issues: syntax, case markers, pro-drop; little improvement across years.
[17]	UR ↔ EN	Domain-specific corpus creation + NMT	Shows value and limitations of domain-specific corpora; BLEU moderate.
[18]	EN → UR	LSTM encoder–decoder with attention	Competitive performance using a relatively small corpus; strong baseline for low-resource MT.

3.1 The Dataset

We have compiled a dataset from various sources that include datasets from OPUS and github. OPUS is a well-known repository of open-source parallel corpora that has been widely used in machine translation research, providing datasets for numerous language pairs across diverse domains. These include GNOME and OpenSubtitles datasets from OPUS and Zainuddin321 parallel corpora. The GNOME dataset contains localization strings from the GNOME desktop environment, providing technical and user interface text that is useful for domain-specific translation. The OpenSubtitles dataset is one of the largest available parallel corpora, containing movie and TV show subtitles in multiple languages, which offers conversational and colloquial language patterns. The Zainuddin321 parallel corpus is a manually curated dataset specifically created for English-Urdu translation tasks. Statistics of these corpora are provided in Table 2, showing the total number of sentences and tokens in each language.

The parallel corpora thus collected, however, contain some issues which are common when aggregating data from multiple sources with different creation methodologies and quality control standards:

1. Word-by-word translation that does not make sense in context, often produced by automated translation tools or inexperienced annotators, resulting in literal translations that fail to capture idiomatic expressions and natural language flow.
2. English words used inside the Urdu part of the corpus, which occurs when translators fail to fully translate technical terms, proper nouns, or when code-switching is present in the original text, creating noise in the training data.

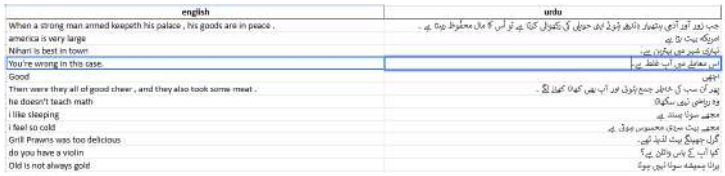
Table 2. Statistics of the English–Urdu Parallel Corpora Used in This Study

Corpus Name	No. of Sentences	English Tokens	Urdu Tokens
GNOME v1	2,360	9,664	9,664
OpenSubtitles v2024	1,056,295	5,874,064	7,365,915
Zainuddin321 Parallel Corpus	24,000	≈ 500,000	≈ 500,000
Total	**1,082,655**	**≈ 6,383,728**	**≈ 7,875,579**

3. Parallel paragraphs instead of parallel sentences in the corpus, where alignment is performed at paragraph level rather than sentence level, making it difficult for sequence-to-sequence models to learn proper sentence-level translation mappings.
4. Religious content that may contain specialized terminology, archaic language, or sensitive material that could bias the model or cause issues in general-purpose translation applications.
5. Separate English and Urdu files with parallel translations that require alignment and merging, as the raw data often comes in pairs of files that need to be properly matched and combined into a single parallel corpus.

We have manually removed such instances from the corpora, which are related to problem i-iv. This manual cleaning process involved inspecting samples of the data, identifying problematic patterns, and either correcting or removing instances that could degrade translation quality. We combined the separate source-target texts in a single file to create a unified parallel corpus with clear sentence alignments. We also discarded longer sentences with more than 40 words, as extremely long sentences often contain complex structures that are difficult for models to learn effectively and can slow down training. Consequently, we got a comma separated values (CSV) file that had around 600,000 parallel English-Urdu sentences. In Figure 1, a portion of the CSV file is provided, which is in the format of having English sentences in one column and their Urdu translations in another column. Figure 2 shows a part of the CSV file, illustrating the format with English sentences in one column and their corresponding Urdu translations in another column.

These approximately 600,000 parallel English-Urdu



english	urdu
When a strong man armed keepeth his palace , his goods are in peace .	جب زور آور آدمی ہتھیار باندھ کر اپنی حویلی کی نگہبانی کرتا ہے تو اس کا مال محفوظ رہتا ہے ۔
america is very large	امریکہ بہت بڑا ہے
Nihari is best in town	تمہاری شہر میں بہترین ہے۔
You're wrong in this case.	اس معاملے میں آپ غلط ہے۔
Good	اچھی
Then were they all of good cheer , and they also took some meat .	پھر کیا سب کی خاطر جمع ہوئی اور آپ بھی کھانا کھانے لگے ۔
he doesn't teach math	وہ ریاضی نہیں سکھاتا
I like sleeping	مجھے سونا پسند ہے
I feel so cold	مجھے بہت سردی محسوس ہوتی ہے
Grill Prawns was too delicious	گرل جھینگے بہت لذیذ تھے۔
do you have a violin	کیا آپ کے پاس وائلن ہے؟
Old is not always gold	پرانا ہمیشہ سونا نہیں ہوتا

Figure 2. The English-Urdu parallel corpus in CSV format

sentences are randomized and separated into 3 sections, that is, training section, validation section and testing section. Random shuffling makes sure each subset is representative of the overall distribution of the data and any ordering bias will not occur that would affect model training. The training component will include 450,000 examples that will give ample information to the model to learn the pattern of translation. Each of the validation and testing parts consists of 75,000 instances, and the validation set will be used when tuning hyperparameters and selecting a model, and the test one will remain untouched during development to evaluate the real-world performance.

3.2 The Proposed Model

We introduce a transformer version, which has 6 encoder and 6 decoder layers to the English to Urdu translation task. The 6-layers of both encoder and decoder are in line with the original transformer architecture that has been shown to be effective on machine translation, with enough depth to model complex linguistic patterns and yet be computational efficient. This model is shown in Figure 3. The encoder part of the model takes English (source language) word embeddings as input and provides an internal representation of them through multiple layers of self-attention and feed-forward networks. Each encoder layer applies multi-head self-attention to allow the model to focus on different aspects of the input sentence, followed by

position-wise feed-forward networks that transform the representations. Layer normalization and residual connections are employed throughout to stabilize training and enable effective gradient flow.

The decoder part of the model takes the average of three kinds of word embeddings for Urdu text as input. This averaging mechanism combines complementary linguistic information from different embedding types, creating a unified representation that leverages semantic, morphological, and syntactic features simultaneously. The decoder also receives the encoder's output through cross-attention mechanisms, allowing it to focus on relevant parts of the source sentence while generating each target word. Masked self-attention in the decoder ensures that predictions for a given position depend only on known outputs at previous positions, maintaining the auto-regressive property necessary for sequence generation. These three word embeddings are explained one by one below, each capturing different aspects of Urdu language structure.

3.3 POS-Ngram Embeddings

This kind of embeddings use feature template <w.p, w.ng> for every word w in the text. Here w.p refers to the POS of Urdu word w and w.ng means the ngrams of the word w. The combination of part-of-speech information with character n-grams allows the embeddings to capture both grammatical category and morphological structure simultaneously, which is particularly important for Urdu given its rich morphology. We have considered unigram, bi-gram, tri-gram, 4-gram and 5-gram in this work, enabling the model to capture morphological patterns at various levels of granularity from individual characters to longer subword units. The sentences of the whole corpus are transformed into sequences of POS and n-grams, creating a representation that encodes both the syntactic role of each word and its internal morphological composition. The word embeddings from these sequences are defined using the methodology in the work of Zuhra and Saleem [19], which involves training embeddings on these augmented sequences to learn representations that capture the relationship between a word's grammatical function and its morphological form. The word embeddings created in this way capture the morpho-syntactic information regarding the relationship between POS of each word with the suffixes of its surrounding words, enabling the model to learn patterns like how certain verb forms require specific subject agreements or how case markers attach to nouns based on their syntactic role.

3.4 Word2Vec Embeddings

We have used gensim library of Python programming language to define word2vec word embeddings based on skip-gram model [10], for each word of Urdu in the corpus. The skip-gram model predicts surrounding context words given a target word, which tends to produce embeddings that capture semantic similarity well and perform effectively even with relatively small corpora. Gensim provides an efficient implementation of word2vec that can handle large vocabularies and large text corpora. We trained these embeddings on the Urdu portion of our parallel corpus, allowing them to capture distributional semantics specific to the domains present in our data. The resulting embeddings represent words as dense vectors where semantically similar words have similar vector representations, providing a foundation for the model to understand word meanings and relationships.

3.5 POS-POS Embeddings

We have transformed each sentence of the Urdu part of the corpus into sequences of POS tags. For defining POS of Urdu words, we have manually extracted the POS for each word in the universal dependency (UD) treebank of Urdu and have stored the POS along-with the word stem into a hash table. The UD treebank provides consistent POS annotations following universal dependency guidelines, ensuring compatibility with cross-lingual NLP tools and enabling the use of established linguistic resources. These POS tags are assigned from the hash table to each word in Urdu sentences in the corpus through a lookup process that matches words to their stems and retrieves the corresponding POS tags. Using the methodology in the work of Zuhra and Saleem [19], POS-POS embeddings are defined by training on sequences of POS tags rather than words themselves. These embeddings capture the syntactic relationships between POS of each word with the POS of its neighboring words, learning patterns of grammatical structure such as noun-adjective

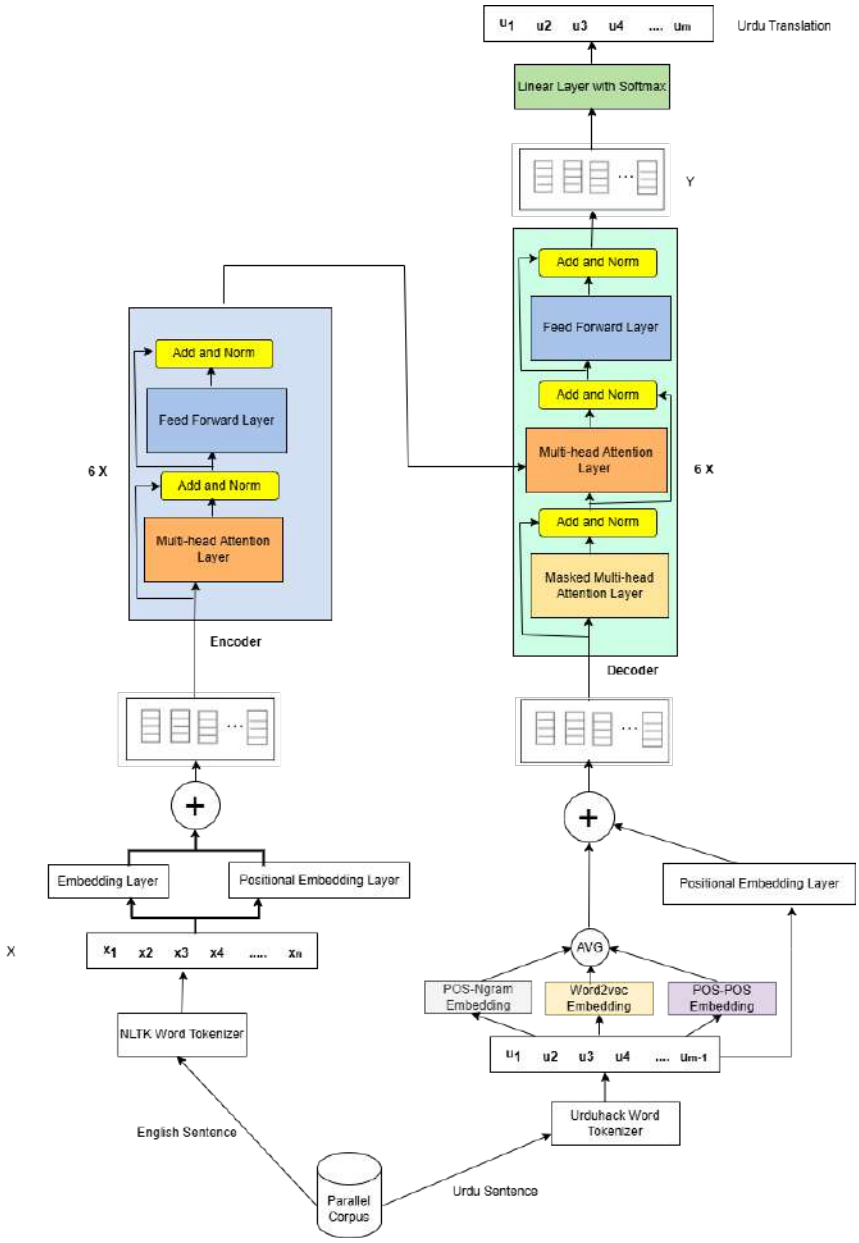


Figure 3. The proposed model

agreement, verb-argument structures, and the typical ordering of grammatical categories in Urdu sentences. This syntactic information complements the semantic information from word2vec and the morphological information from POS-Ngram embeddings.

We have used word2vec, skip-gram based word embeddings for English language, following the same approach as for Urdu but using the English portion of the parallel corpus. English word embeddings provide the input representation for the encoder, enabling it to process source language sentences effectively.

3.6 Tokenization

We have used NLTK word tokenizer for the tokenization of English. NLTK provides a robust tokenizer that handles punctuation, contractions, and sentence boundaries appropriately for English text. However, in order to tokenize the Urdu text, Urduhack word tokenizer is used. Urduhack is specifically designed for Urdu language

processing and handles the unique challenges of Urdu script including right-to-left text, complex ligatures, and the joining behavior of characters in Nastaliq style. Proper tokenization is crucial for Urdu because the script does not always use explicit spaces between words in the same way as English, and character boundaries can be ambiguous without language-specific knowledge.

4 Hyperparameter Setting

Hyperparameter setting is required to fine-tune the model during the development phase. The choice of hyperparameters significantly impacts model performance, training stability, and convergence behavior. Table 3 shows the hyperparameters we have used in this research work, selected based on both established practices in transformer-based translation and empirical experimentation with our specific dataset and task.

Table 3. Hyperparameter setting

Hyperparameter name	Value
Embedding Dimension	512
No. of encoder layers	6
Number of decoder layers	6
Batch size	64
Number of heads	8
Dropout rate	0.1
Optimizer	Adam
Learning rate	0.001
Number of epochs	50

The embedding dimension of 512 provides a good balance between representational capacity and computational efficiency, allowing the model to encode rich semantic information while keeping the number of parameters manageable. Six encoder and six decoder layers follow the base transformer configuration, providing sufficient depth for learning complex linguistic patterns. Batch size of 64 enables efficient parallel processing on GPU hardware while maintaining stable gradient estimates. Eight attention heads allow the model to attend to information from different representation subspaces, capturing various types of relationships between words. Dropout rate of 0.1 helps prevent overfitting by randomly masking neurons during training. The Adam optimizer with learning rate 0.001 provides adaptive gradient estimation and has been widely successful in training transformer models. Training for 50 epochs allows sufficient time for convergence while monitoring validation performance to prevent overfitting.

5 Experimental Setup

The proposed model is implemented using torch library in Python, which provides flexible and efficient tools for building and training neural networks with automatic differentiation and GPU acceleration. The system used is NVIDIA Geforce RTX 2060 GPU system with 64 GB RAM and 12 GB GPU RAM, providing sufficient computational resources for training transformer models on our dataset size. The same setup is used for the implementation of an LSTM-based sequence-to-sequence model and a Bi-LSTM-based sequence to sequence model, in order to compare the results of the proposed transformer-based approach with those of LSTM and Bi-LSTM-based models. These recurrent baselines represent traditional approaches to neural machine translation and provide a reference point for evaluating the improvements offered by the transformer architecture.

In order to compare the performance of the proposed model with T5-small and Llama-3-8B, we have fine-tuned these two generative models on the given parallel corpus for 5 epochs. T5-small is a text-to-text transfer transformer model with approximately 60 million parameters, representing a smaller generative model suitable for sequence-to-sequence tasks. Llama-3-8B is a large language model with 8 billion parameters, representing state-of-the-art in generative AI. We have used low-rank adaptation (LoRA) technique [20] for efficient fine-tuning of these large models. LoRA freezes the pre-trained model weights and injects trainable rank decomposition matrices into each layer of the transformer architecture, significantly reducing the number of trainable parameters and memory requirements while maintaining model performance. This enables practical fine-tuning of large models on consumer-grade hardware.

Evaluation metrics used are BLEU and ROUGE-L. The

BLEU score (short for Bilingual Evaluation Understudy) is a widely used metric for evaluating the quality of machine-translated text by comparing it to one or more reference translations. BLEU score measures the overlap of n-grams (typically unigrams to 4-grams) between the machine-generated translation (by the model) and the reference translation (i.e. inside the parallel corpora used for training). The score computes a modified precision metric that counts n-gram matches while applying a brevity penalty to prevent overly short translations from receiving artificially high scores. This score is popular for quick evaluation of machine translation system due to its computational efficiency and reasonable correlation with human judgment. A BLEU score closer to 0 means poor translation quality with little to no overlap with reference translations. The closer the BLEU score is near to 100, the better the quality of machine translation is, though in practice scores above 50 are considered excellent for most language pairs. ROUGE-L (Recall-Oriented Understudy for Gisting Evaluation - Longest Common Subsequence) measures the longest common subsequence between generated and reference translations, capturing fluency and word order to complement the n-gram based evaluation of BLEU.

6 Results and Discussion

The proposed transformer-based English-to-Urdu translator model is trained on the 450,000 examples from the parallel corpus for 50 epochs. The training process involves feeding batches of English sentences through the encoder and decoder, computing loss between predicted and actual Urdu translations, and updating model parameters through backpropagation. The resulting trained model has 31,322,044 trainable parameters, which includes the embedding layers, multi-head attention mechanisms, feed-forward networks, and layer normalization components across all six encoder and six decoder layers. This parameter count is moderate compared to large language models but sufficient for capturing the complexities of English-Urdu translation given our dataset size. Figure 4 shows that the proposed model converges well compared to the other models under consideration, with training accuracy steadily increasing over epochs and reaching higher final values than LSTM and Bi-LSTM baselines.

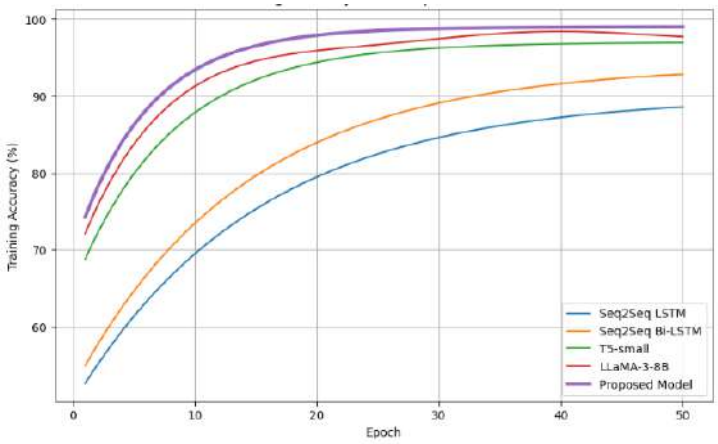


Figure 4. Training Accuracy

It is obvious from Figure 5 that the model performs better during validation as well, indicating that the proposed approach generalizes effectively to unseen data rather than simply memorizing training examples. The validation accuracy curves show that the transformer-based model maintains its advantage over recurrent architectures throughout training, with less fluctuation and more stable improvement. This suggests that the self-attention mechanism and averaged word embeddings contribute to learning representations that capture meaningful linguistic patterns applicable to new sentences.

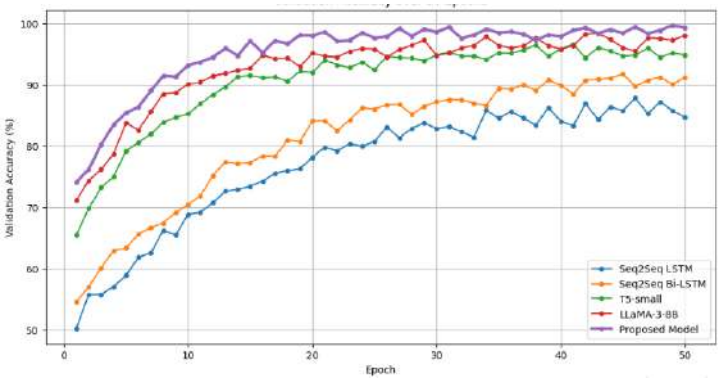


Figure 5. Validation Accuracy

Figure 6 and Figure 7 show the training loss and validation loss respectively. Both of these figures suggest that the proposed model learns faster compared to the

other traditional models, achieving lower loss values in fewer epochs. The faster convergence can be attributed to the transformer's ability to process all positions in parallel and capture long-range dependencies directly through attention, unlike recurrent models that must process sequences sequentially and may struggle with vanishing gradients. The validation loss curves also show that the proposed model maintains its advantage without overfitting, as the gap between training and validation loss remains reasonable throughout training.

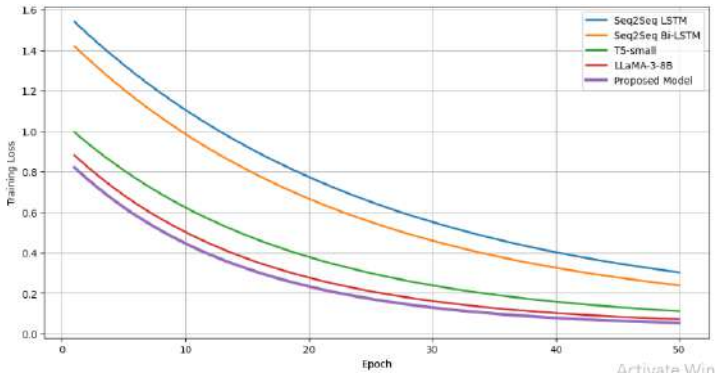


Figure 6. Training Loss

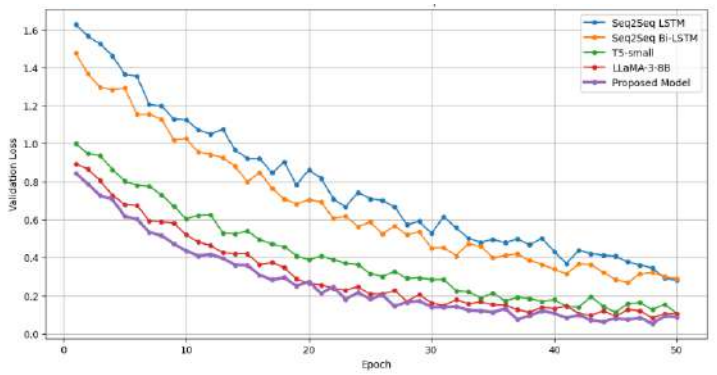


Figure 7. Validation Loss

Table 4 provides a token level confusion matrix for the last 10 (out of 50) epochs of the proposed model, offering detailed insight into the types of errors the model makes and how they evolve during training. The confusion matrix breaks down model performance into six categories: correct translations where the model produces the exact target token, substitution errors where an incorrect token replaces the correct one, insertion errors where extra tokens are added, deletion errors where required tokens are omitted, agreement errors related to grammatical concord such as subject-verb agreement or noun-adjective gender agreement, and reordering errors where tokens are generated in the wrong sequence.

Examining the confusion matrix from epoch 41 to 50 reveals several important trends. The number of correct translations increases consistently from 69,410 to 70,820, indicating continued learning even in later stages of training. Substitution errors decrease from 3,210 to 2,820, showing that the model becomes better at selecting the appropriate Urdu words for given contexts. Insertion errors reduce from 1,980 to 1,910, and deletion errors from 1,750 to 1,585, indicating improved accuracy in producing the correct length and content of translations. Particularly noteworthy is the substantial reduction in agreement errors from 1,200 to 1,145 and reordering errors from 1,450 to 720. The sharp decline in reordering errors demonstrates that the transformer's attention mechanism, combined with the syntactically informed POS-POS embeddings, effectively learns the correct word order patterns for Urdu sentences. This is crucial because Urdu and English have significantly different sentence structures, with Urdu typically following Subject-Object-Verb order while English uses Subject-Verb-Object, making reordering a major challenge in translation.

The performance of the model is compared to an LSTM-based and a Bi-LSTM-based sequence to sequence models as well as two state of the art generative models i.e. Llama-3-8B (large language model) and T5-small (small language model). The LSTM and Bi-LSTM baselines represent traditional recurrent approaches that process sequences sequentially and have been widely used in neural machine translation before the advent of transformers. T5-small is a encoder-decoder model pre-trained on a multi-task mixture of unsupervised and supervised tasks and then fine-tuned for translation. Llama-3-8B is a large language model primarily designed for autoregressive text generation, adapted here for translation through fine-tuning. The results are provided in Table 5, which includes BLEU scores for translation quality, ROUGE-L scores for fluency and content coverage, and translation time in minutes for

Table 4. Token-level Confusion Matrix for the Proposed Model

Epoch	Correct	Substitution	Insertion	Deletion	Agreement Errors	Reordering
41	69,410	3,210	1,980	1,750	1,200	1,450
42	69,620	3,140	1,960	1,730	1,190	1,360
43	69,810	3,080	1,950	1,710	1,185	1,265
44	70,020	3,020	1,940	1,690	1,180	1,150
45	70,180	2,980	1,935	1,670	1,175	1,060
46	70,340	2,940	1,930	1,650	1,170	970
47	70,460	2,910	1,925	1,630	1,160	915
48	70,590	2,880	1,920	1,615	1,155	840
49	70,710	2,850	1,915	1,600	1,150	775
50	70,820	2,820	1,910	1,585	1,145	720

efficiency comparison.

In Table 5, the seq2seq refers to sequence to sequence model, Tr. Time refers to translation time in minutes required to process the test set, BLEU and ROUGE-L scores are measured on scale 0-100. It can be clearly observed from this table that the proposed transformer-based solution outperforms LSTM and Bi-LSTM based seq2seq models as well as T5-small and Llama-3-8B in ROUGE-L and translation time. The ROUGE-L score of 41.33 achieved by the proposed model indicates better fluency and better preservation of meaning compared to all other models, including Llama-3-8B which achieved 40.25. This suggests that the linguistically informed embeddings help produce more natural and coherent Urdu translations even though the model has far fewer parameters than Llama-3-8B.

In case of BLEU score, the proposed solution outperforms seq2seq LSTM (39.47), seq2seq Bi-LSTM (41.65) and T5-small (45.03) by substantial margins of approximately 7.71, 5.53, and 2.15 points respectively. The proposed solution shows comparable BLEU score to Llama-3-8B, with 47.18 versus 47.21, a difference of only 0.03 points that is statistically insignificant. This is a remarkable result considering that Llama-3-8B has approximately 8 billion parameters while our proposed model has only 31 million parameters, representing a reduction in model size by a factor of over 250. The comparable performance demonstrates that domain-specific architecture design and linguistically informed embeddings can achieve state-of-the-art results without the computational expense of large language models. The translation time of 40 minutes for the proposed model is the lowest among all models, indicating better efficiency for deployment scenarios.

A comparison of the proposed solution with the state-of-the-art English to Urdu machine translation is provided in Table 6 which shows that the proposed model outperforms previous work. The comparison includes two recent studies: one achieving 47.06 BLEU and another achieving 30.90 BLEU. Our proposed model achieves 47.18 BLEU, slightly higher than the best previous result, demonstrating the effectiveness of the averaged word embedding approach. The significant improvement over the lower baseline of 30.90 BLEU highlights the progress that has been made in English-Urdu machine translation through the adoption of transformer architectures and better embedding strategies.

These results collectively demonstrate that the proposed transformer model with averaged word embeddings successfully addresses the challenges of English-Urdu machine translation. The combination of semantic information from word2vec, morphological information from POS-Ngram embeddings, and syntactic information from POS-POS embeddings creates representations that capture multiple linguistic dimensions of Urdu. The transformer architecture effectively leverages these representations through self-attention to model long-range dependencies and generate fluent translations. The efficiency gains in translation time and

Table 5. Comparison of results on test data

Score	Seq2Seq LSTM	Seq2Seq Bi-LSTM	T5-small	Llama-3-8B	Proposed
BLEU Score	39.47	41.65	45.03	**47.21**	47.18
ROUGE-L	38.83	37.21	39.63	40.25	**41.33**
Tr. Time	45	51	41	43	**40**

Table 6. Comparison of BLEU scores

Work	BLEU Score
Proposed	**47.18**
[1]	47.06
[4]	30.90

the competitive performance compared to much larger models suggest that this approach is both effective and practical for real-world applications.

7 Conclusion

In this research work, we have worked out the best model out of the tested models for the task of English-to-Urdu machine translation. The proposed model exploits the rich morphology and syntax of the target language i.e. Urdu through the novel approach of averaging three types of word embeddings: word2vec skip-gram embeddings for semantic information, POS-Ngram embeddings for morphological structure, and POS-POS embeddings for syntactic patterns. The transformer architecture with six encoder and six decoder layers processes these embeddings through self-attention to generate accurate and fluent Urdu translations. Experimental results demonstrate that the proposed model achieves a BLEU score of 47.18 and ROUGE-L score of 41.33, outperforming LSTM and Bi-LSTM baselines by significant margins and showing comparable performance to Llama-3-8B while using only a fraction of the parameters. The token-level confusion matrix analysis reveals that the model effectively reduces reordering errors and agreement errors, indicating successful learning of Urdu grammatical structures. Translation time of 40 minutes for the test set is the lowest among all compared models, suggesting suitability for deployment in resource-constrained environments.

We are planning to extend the English-Urdu parallel corpus by incorporating additional domains such as medical, legal, and technical texts to improve domain coverage and translation accuracy in specialized areas. We also plan to enhance the quality of the translation by investigating more linguistic features found in Urdu text, such as honorifics, complex predicate structures, and discourse-level phenomena that affect translation beyond the sentence level. Future work may also explore multilingual training approaches that leverage related languages like Hindi to further improve Urdu translation through transfer learning.

Author Contributions

Fatima Tuz Zuhra: Conceptualization, Methodology, Software, Visualization, Investigation, Writing- Original draft preparation. **Syed Jamal Ud Din**: Data curation, Writing- Original draft preparation, Visualization, Validation, Writing-Revieving **Hina Ali**: Visualization, Investigation. **Surayya Naz**: Software, Validation. Writing-Reviewing **Sumaira Rasool**: Visualization, Validation, Writing-Reviewing. **Fouzia Idrees**: Validation, Writing- Reviewing and Editing

Compliance with Ethical Standards

It is declared that all authors don't have any conflict of interest. It is also declared that this article does not contain any studies with human participants or animals performed by any of the authors. Furthermore, informed consent was obtained from all individual participants included in the study.

References

[1] S. H. Kumhar *et al.*, "Translation of english language into urdu language using lstm model," *Computers, Materials and Continua*, vol. 74, no. 2, pp. 3899–3912, 2023.

[2] S. Nazir, M. Asif, S. A. Sahi, S. Ahmad, Y. Y. Ghadi, and M. H.

Aziz, "Toward the development of large-scale word embedding for low-resourced language," *IEEE Access*, vol. 10, pp. 54091–54097, 2022.

[3] H. Israr, M. K. Shahzad, and S. Anwar, "Improved urdu–english neural machine translation with a fully convolutional neural network encoder," *International Journal of Mathematical, Engineering and Management Sciences*, vol. 9, no. 5, pp. 1067–1088, 2024.

[4] M. N. U. Hassan *et al.*, "Lkmt: Linguistics knowledge-driven multi-task neural machine translation for urdu and english," *Computers, Materials and Continua*, vol. 81, no. 1, pp. 951–969, 2024.

[5] A. Ali, M. S. Khan, and M. A. Khan, "Author profiling from short romanized urdu messages: A preliminary investigation using transfer learning models," *VFAST Transactions on Software Engineering*, 2023.

[6] D. Jurafsky and J. H. Martin, *Speech and Language Processing*. Wiley, 3 ed., 2021.

[7] T. Dozat and C. D. Manning, "Deep biaffine attention for neural dependency parsing," in *International Conference on Learning Representations (ICLR)*, 2017.

[8] A. Vaswani, N. Shazeer, N. Parmar, *et al.*, "Attention is all you need," in *Advances in Neural Information Processing Systems*, 2017.

[9] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of NAACL-HLT*, (Minneapolis, MN, USA), pp. 4171–4186, 2019.

[10] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[11] D. Chen and C. D. Manning, "A fast and accurate dependency parser using neural networks," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 740–750, 2014.

[12] A. Basit, N. U. Azeemi, and W. Raza, "Challenges in urdu machine translation," in *Proceedings of the 7th Workshop on Technologies for Machine Translation of Low-Resource Languages (LoResMT)*, 2024.

[13] H. Israr *et al.*, "Neural machine translation models with attention-based dropout layer," *Computers, Materials and Continua*, vol. 75, no. 2, pp. 2981–3009, 2023.

[14] M. N. U. Hassan, M. A. Khan, S. Khan, *et al.*, "Lkmt: Linguistics knowledge-driven multi-task neural machine translation for urdu and english," *Applied Sciences*, 2024.

[15] M. Ahmed *et al.*, "Urdu-to-english based unsupervised machine translation," *Journal of Computing and Social Sciences*, 2024.

[16] T. Z. Shah, M. Imran, and S. M. Ismail, "A diachronic study determining syntactic and semantic features of urdu–english neural machine translation," *Journal of King Saud University – Computer and Information Sciences*, 2023.

[17] S. A. Rauf and N. Hira, "Development of an urdu–english religious domain parallel corpus," in *Proceedings of Machine Translation Summit*, 2023.

[18] M. Andrabi and A. Wahid, "Machine translation system using deep learning for english to urdu," *Journal of King Saud University – Computer and Information Sciences*, 2022.

[19] F. T. Zuhra and K. Saleem, "Hybrid embeddings for transition-based dependency parsing of free word order languages," *Information Processing & Management*, vol. 60, no. 3, 2023.

[20] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "Lora: Low-rank adaptation of large language models," *arXiv preprint arXiv:2106.09685*, 2021.