JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

Neural Machine Translation for Under-Represented Indian Languages

Khush Fadadu[1] , Charmi Kalyani[2] , Hirak Modi[3] , Madhvi Bera[4]

1. B-tech Student, Computer Science and Engineering, Indus Institute of Technology and Engineering, Ahmedabad – 382115
2. B-tech Student, Computer Science and Engineering, Indus Institute of Technology and Engineering, Ahmedabad – 382115
3. B-tech Student, Computer Science and Engineering, Indus Institute of Technology and Engineering, Ahmedabad – 382115
4. Professor , Computer Science and Engineering, Indus Institute of Technology and Engineering, Ahmedabad – 382115

Abstract :- Many useful things are available on the internet in English, but not everyone understands English well. So, we often need to translate these things into local languages to help people who don't speak English. But doing this translation by hand is hard, expensive, and takes a long time. That's where machine translation comes in – it's a way for computers to automatically change text from one language to another without people having to do it. Among the different ways computers can do this, there's one called neural machine translation (NMT), which is really good at it. In this research paper, we talk about using NMT for two complicated Indian languages: English-Tamil and English-Malayalam. These languages are tricky because they have lots of unique features, and there aren't many online tools to help with translations. To make this work, we came up with a new way of using NMT. We added something called Multihead self-attention and used special pre-trained techniques called Byte-Pair-Encoded (BPE) and MultiBPE embeddings. These fancy words might sound complicated, but they help our system handle words it doesn't know well (Out Of Vocabulary or OOV words). We also collected texts from different places, fixed problems in the data we found online, and made it all better for our system to use. To check how well our system worked, we used something called the BLEU score. And guess what? Our system did really well! It got high scores of 24.34 for English-Tamil and 9.78 for English-Malayalam translations. That's better than what Google Translator could do, which got scores of 9.40 and 5.94 for the same translations.

KEYWORDS
Multihead self-attention, Byte-Pair-Encodding, MultiBPE, low-resourced, Morphology, Indian Languages

1. Introduction

Many heavily populated countries like India and China exhibit significant linguistic diversity, with multiple languages varying from region to region. For instance, India officially recognizes 23 languages, including Hindi, Malayalam, Telugu, Tamil, and Punjabi, along with numerous unofficial local languages. Even smaller nations can boast rich language diversity, as seen in Papua New Guinea, where 851 languages are spoken, despite its relatively low population.In India, where the population is approximately three billion, only around 10% of the populace can speak English. Moreover, studies suggest that out of this 10%, only 2% are proficient in reading, writing, and comprehending English effectively, while the remaining 8% can merely grasp basic English and communicate with diverse accents. Given that a wealth of valuable information is available on the internet in

English, and a significant portion of India's population struggles to understand it comprehensively, the need to translate such content into local languages becomes paramount. Facilitating the exchange of information among people is essential not only for business purposes but also for sharing emotions, opinions, and actions. In this context, translation plays a crucial role in bridging the communication gap between various communities. Considering the vast volume of text available, manual translation is impractical. Hence, the automatic translation of text from one language, such as English, into other languages like Tamil and Malayalam, becomes essential. This approach is commonly referred to as machine translation.

Indian languages exhibit substantial differences compared to English, particularly in terms of their morphological richness and variations in word order due to syntactical differences. Indian languages, such as Malayalam and Tamil, not only differ in word order but are also more agglutinative compared to English, which is fusional. For example, English follows a Subject-Verb-Object (SVO) word order, while Tamil and Malayalam follow a Subject-Object-Verb (SOV) word order.These syntactic and morphological distinctions present significant challenges for translation models. Syntactic differences complicate the task of translation, while morphological differences exacerbate data sparsity issues. In this paper, we aim to address both of these challenges.

Many research papers in the field of machine translation predominantly focus on foreign languages, with a significant emphasis on languages like Hindi. They often employ traditional machine translation techniques as evidenced by works such as (Patel et al., 2018) and (Raju and Raju, 2016). Much of the prior research has centered around the segmentation of words into suffixes and prefixes based on certain rules, followed by the application of translation methods. In our work, we address this limitation by leveraging Byte-Pair-Encoding (BPE), a technique that enhances the efficiency and reliability of the entire translation process.

Furthermore, it's important to note that there has been limited exploration of low-resource Indian languages, and innovative techniques such as Byte-Pair-Embedding (BPEmb), MultiBPEmb, word embeddings, and self-attention mechanisms have not received due attention. These techniques have demonstrated significant improvements in Natural Language Processing but have been underutilized in the context of Indian languages. While unsupervised machine translation, as explored in (Artetxe et al., 2017), has garnered some attention, it still lacks the precision of supervised learning.

Another significant challenge we identify is the scarcity of trustworthy public data for translating these languages. To address these issues, we propose a neural machine translation approach that incorporates Multihead self-attention, word embeddings, and Pre-Trained Byte-Pair-Encoding. We specifically focus on the challenging English-Tamil and English-Malayalam language pairs, given their morphological richness, which makes them particularly difficult to translate.

It's important to highlight that our approach can be adapted to other languages as well. To support our research, we gathered data from En-Tamv2.0, Opus, and UMC005, meticulously preprocessed it, and evaluated our results using the BLEU evaluation metric. The implementation of our models was carried out using OpenNMT-py.

The experimental results, as well as feedback from native speakers, confirm that our approach outperforms conventional translation techniques when applied to Indian languages. Our work's main contributions can be summarized as follows:

We are the first to apply pre-trained BPE and MultiBPE embeddings to Indian language pairs, particularly English-Tamil and English-Malayalam, in conjunction with the Multihead self-attention technique. We achieve commendable accuracy with a relatively straightforward model and a shorter training time, as opposed to more resource-intensive and time-consuming complex neural networks. We address the critical issue of data preprocessing for Indian languages, underscoring its significance in the context of neural machine translation.

We make our preprocessed data publicly available, potentially comprising the most extensive parallel corpus for languages such as English-Tamil, English-Malayalam, English-Telugu, English-Bengali, and English-Urdu, a valuable resource for the research community.

Our model outperforms Google Translator with a notable margin, boasting an 18.07 BLEU score.

2. Background

Machine translation (MT) has been a subject of extensive research spanning several decades, with its origins dating back to the 1950s, as exemplified by Booth (1955). Researchers have explored a variety of approaches in the field, including rule-based (RBMT), corpus-based, and hybrid-based methods. Each approach has its own set of strengths and weaknesses.

Rule-based machine translation (RBMT) relies on linguistic information about the source and target languages obtained from multilingual, bilingual, or monolingual dictionaries and grammars. It encompasses two major subcategories: transfer-based (TBA) and inter-lingual-based (IBA) approaches. TBA, as exemplified by Shilon (2011), involves transferring linguistic elements from the source to the target language, while IBA explores interlingual representations of the source and target languages.

In the corpus-based approach, large parallel corpora, containing ground-truth translations for the desired languages, serve as the raw data for training translation models. This approach further divides into statistical machine translation (SMT), as seen in Patel et al. (2018), and example-based machine translation (EBMT), as discussed by Somers (2003). SMT combines decoding algorithms with statistical language models, while EBMT relies on translation examples to generate new translations by identifying matching examples and aligning their components for reuse.

Hybrid-based machine translation seeks to overcome the limitations of both corpus-based and transfer-based approaches by combining elements of both. Recent research, such as Khan et al. (2017), indicates that machine translation performance for Indian languages, including Hindi, Bengali, Tamil, Punjabi, Gujarati, and Urdu, hovers at an average of 10% accuracy. This highlights the pressing need for more effective translation systems for Indian languages.

Unsupervised machine translation is an emerging paradigm that aims to translate without relying on parallel corpora, but the results have not yet been notably remarkable. In contrast, Neural Machine Translation (NMT) has emerged as a promising technique that has demonstrated significant improvements in translation quality. In the work of Hans and Milton (2016), a phrase-based hierarchical model was used, which was trained after morphological preprocessing. Patel et al. (2017) trained their model after compound splitting and suffix separation. Many other researchers have pursued similar approaches and achieved respectable results on their respective datasets (e.g., Pathak and Pakray). Our observation suggests that the challenges posed by morphological preprocessing, compound splitting, and suffix or prefix separation can be effectively addressed by employing Byte-Pair-Encoding (BPE), which can yield similar or even superior translation results without introducing unnecessary model complexity.

3. Approach

In this paper, we introduce a novel approach to neural machine translation (NMT) that incorporates Multihead self-attention, word embeddings, and pre-trained Byte-Pair-Encoding (BPE) on our meticulously preprocessed dataset of Indian languages. Our objective is to develop an efficient translation system capable of addressing challenges such as Out Of Vocabulary (OOV) terms and the intricacies of morphological analysis, especially for Indian languages with limited online translation resources. To achieve this, we first provide an overview of NMT, Multihead self-attention, word embedding, and Byte Pair Encoding. Subsequently, we delve into the framework of our translation model.

3.1 Neural Machine Translation Overview

Neural Machine Translation (NMT) represents a robust algorithm based on neural networks that leverages conditional probabilities to predict target sentences from source language sentences (Revanuru et al., 2017a). When combined with attention mechanisms, this architecture can yield impressive results across various language pairs. The following subsections offer an overview of the fundamental aspects of NMT, including sequence-to-sequence architecture, self-attention, and other techniques that are integral to our proposed translation model.

3.1.1. Sequence to sequence architecture

Sequence to sequence architecture is used for response gen-eration whereas in Machine Translation systems it is used to find the relations between two language pairs. It con-sists of two important parts, an encoder, and a decoder. Theencoder takes the input from the source language and the decoder leads to the output based on hidden layers and pre-viously generated vectors. Let A be the source and B be a target sentence. The encoding part converts the source sen-tence a_1, a_2, a_3..., a_n into the vector of fixed dimensions and the decoder part gives the word by word output using conditional probability. Here, A_1, A_2, ..., A_M in the equa- tion are the fixed size encoding vectors. Using chain rule, the Eq. 1 is transformed to the Eq. 2.

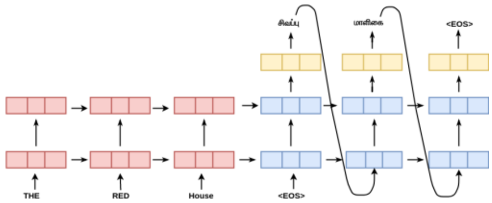

Figure 1: Seq2Seq architecture for English-Tamil

$$P(B/A) = P(B|A_1, A_2, A_3, ..., A_M) \qquad (1)$$

$$P(B|A) = P(b_i|b_0, b_1, b_2, ..., b_{i-1}; \qquad (2)$$
$$a_1, a_2, a_3, ..., a_m$$

The decoder generates output using previously predicted word vectors and source sentence vectors in Eq. 1.

3.1.2 Attention Model

In a fundamental encoder-decoder architecture, the encoder processes the entire sentence and stores it as a vector in the final activation layer. This vector is then utilized by the decoder to generate the target sentence. While this architecture performs reasonably well for shorter sentences, for longer sentences, typically exceeding 30 or 40 words, its performance tends to degrade. To address this challenge, attention mechanisms come into play. The underlying concept is that when the model predicts an output word, it should focus on the parts of the input sentence where the most relevant information is concentrated, rather than considering the entire sentence. In other words, it should selectively attend to weighted words. Various attention mechanisms have been developed to enhance translation accuracy, and the multi-head self-attention mechanism stands out as a promising solution.

Self-Attention: The self-attention architecture, as introduced by Vaswani et al. (2017), involves calculating a weighted average of all previous states at every time step of a Recurrent Neural Network (RNN). This weighted average is used as an additional input for determining the next state. With self-attention, the network can decide to pay attention to a state produced many time steps earlier, reducing the reliance on the latest state to store all the information. This mechanism also facilitates the flow of gradients to all previous states, mitigating issues related to the vanishing gradient problem.

Multi-Head Attention: When we have multiple queries (q), they can be combined into a matrix Q. If we compute alignment using dot-product attention, the equations used to calculate context vectors can be streamlined, as depicted in Figure 3. Q, K (keys), and V (values) are mapped into lower-dimensional vector spaces through the use of weight matrices. The results of these transformations are then employed to compute attention, often referred to as a "Head." This multi-head approach allows for enhanced modeling of complex relationships between different elements in the input and output sequences, offering improved translation accuracy and capturing more context in parallel.

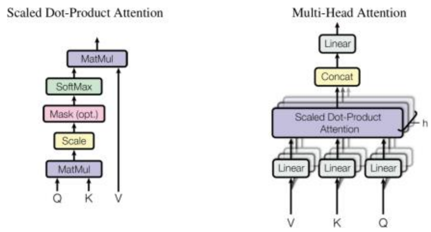

Figure 2: Attention model

In Multi-Head Attention we have h such sets of weight matrices which give us h Heads.

$$A(q, K, V) = \sum_i \frac{e^{q \cdot k_i}}{\sum_j e^{q \cdot k_j}} v_i$$

$$A(Q, K, V) = softmax(QK^T)V$$

Figure 3: Multi-Head Attention

3.1.3 Word Embedding

Word embedding is a unique method for representing words in a vector space, enabling the capture of semantic similarities between words. Each word is represented in a high-dimensional vector space, typically spanning hundreds of dimensions. In practice, pre-trained embeddings, often trained on large datasets, are employed. Through the use of transfer learning, words from the vocabulary are transformed into vectors, as demonstrated by Cho et al. (2014).

3.1.4 Byte Pair Encoding

Byte Pair Encoding (BPE), introduced by Gage in 1994, is a data compression technique that focuses on replacing the most frequent pairs of bytes in a sequence. In the context of our work, BPE serves a crucial role in word segmentation. By merging frequently occurring pairs of characters or character sequences, BPE enables the creation of a vocabulary of a desired size, as shown by Sennrich et al. (2015). BPE is particularly valuable for tasks involving suffix and prefix separation, as well as compound splitting. In the case of languages like Malayalam and Tamil, BPE helps in generating new and complex words by interpreting them as sub-word units. In our model, we have effectively integrated BPE with pre-trained fastText word embeddings (Heinzerling and Strube, 2018) for both languages, adjusting the vocabulary size to 25,000 and the dimension to 300, which yielded the best results.

MultiBPEmb : MultiBPEmb is a collection of subword segmentation models and pre-trained subword embeddings for multiple languages. These embeddings are trained on Wikipedia data, akin to monolingual BPE, but with a unique twist. Instead of training separate segmentation models for each language, a single model and a single embedding are developed for all the languages. This approach is particularly beneficial for handling mixed-language sentences, including native languages alongside English, which have gained popularity, especially on social media platforms. Given that our sentences were clean and well-structured, MultiBPEmb yielded consistent results with minor variations in BLEU scores, notably increasing translation quality by 0.60 in Tamil and 1.15 in Malayalam.

ID	Language	Train	Test	Dev
1	Tamil	183451	2000	1000
2	Malayalam	548000	3660	3000
3	Telugu	75000	3897	3000
4	Bengali	658000	3255	3500
5	Urdu	36000	2454	2000

Table 1: Dataset for Indian Languages

4 Experimentation and Results

In our experimentation, we addressed several critical issues related to translation quality, tokenization of Indian languages, and dataset preprocessing. To ensure reliable and meaningful results, we implemented various techniques and improvements.

4.1 Evaluation Metric

We measured translation quality using the BLEU score, which is a widely used metric for comparing machine translation output to human translations (Papineni et al., 2002). BLEU score works by matching n-grams in the translation output with those in reference texts. Unigrams represent individual tokens, bigrams consist of pairs of words, and so on. A perfect match results in a BLEU score of 1.0 or 100%.

4.2 Dataset

Our dataset was compiled from diverse sources, including EnTamV2.0 (Ramasamy et al., 2012), Opus (Tiedemann, 2012), and UMC005 (Jawaid and Zeman, 2011). It contains sentences from various domains such as news, cinema, the Bible, and movie subtitles. This dataset covers Tamil, Malayalam, Telugu, Bengali, and Urdu languages. After meticulous preprocessing and cleaning, we divided the dataset into training, testing, and validation sets. To the best of our knowledge, this dataset stands as the largest, clean, and preprocessed public resource available on the web, suitable for a wide range of applications. Given the lack of publicly available datasets for Indian languages, our dataset can serve as a baseline for future comparisons and research.

4.3 Data Pre-processing

We encountered several critical issues while working with publicly available corpora. These issues included repeated sentences with the same source and target, causing potential bias and overfitting, especially when dividing the data into training, validation, and test sets. Additionally, we noted that existing tokenization libraries designed for the English language did not perform well on Indian languages due to significant morphological differences. Indian languages have unique word formations that can best be handled by specific libraries or Byte-Pair-Encoding (BPE). BPE simplifies the tokenization process and often results in more effective translation.

To address these issues, we applied minor but highly effective data preprocessing techniques. We eliminated sentences with lengths exceeding 50 words, sentences with known translated words in the target sentences, noisy translations, and unnecessary punctuation. For data reliability, we also enlisted the assistance of native speakers of the languages.

4.4 Translator

We experimented with various techniques to enhance the translation quality for the Indian language pairs. Our first model included a 4-layer Bi-directional LSTM encoder and a decoder with 500 dimensions each, along with a vocabulary size of 50,004 words for both the source and target languages. We initially used Bahdanau's attention mechanism, the Adam optimizer, a dropout rate of 0.3 for regularization, and a learning rate of 0.001. In this model, we utilized 300-dimensional pre-trained fastText word embeddings for both languages.

In our second model, we introduced pre-trained fastText Byte-Pair-Encoding (BPE) with the same attention mechanism. The third model replaced the attention mechanism with multi-head attention comprising 8 heads and 6 encoding and decoding layers. This change resulted in notable improvements, with a 1.2 BLEU score increase for Tamil and a 6.18 BLEU score increase for Malayalam.

For the final model, we used Multilingual fastText pre-trained Byte-Pair-Encodings, which yielded the best results, with 9.67 and 25.36 BLEU scores for Tamil and Malayalam, respectively, as shown in Table 2 and Table 3.

4.5 Results

Our results, as presented in Table 2 and Table 3, were compared with translations obtained using Google Translate through a Python API. The tests demonstrated that our model effectively addressed Out of Vocabulary (OOV) problems in some cases, making it a practical tool for daily and official use.

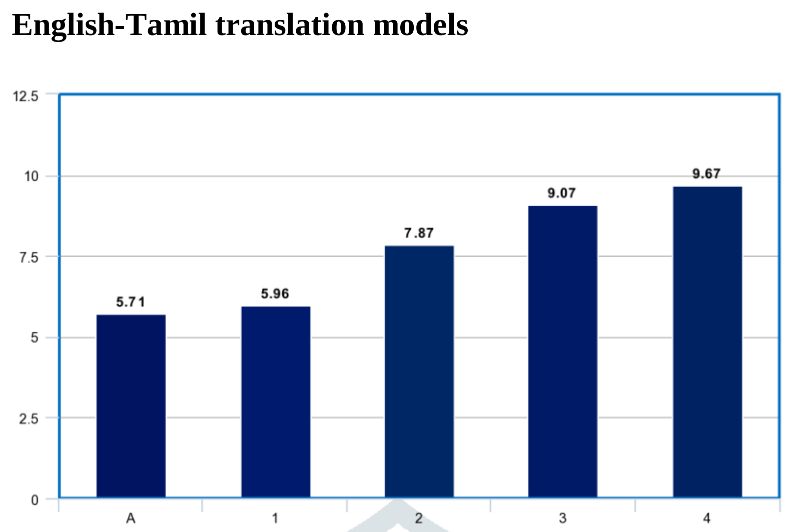

Figure 4: English-Tamil model comparison with Google Translator Table2

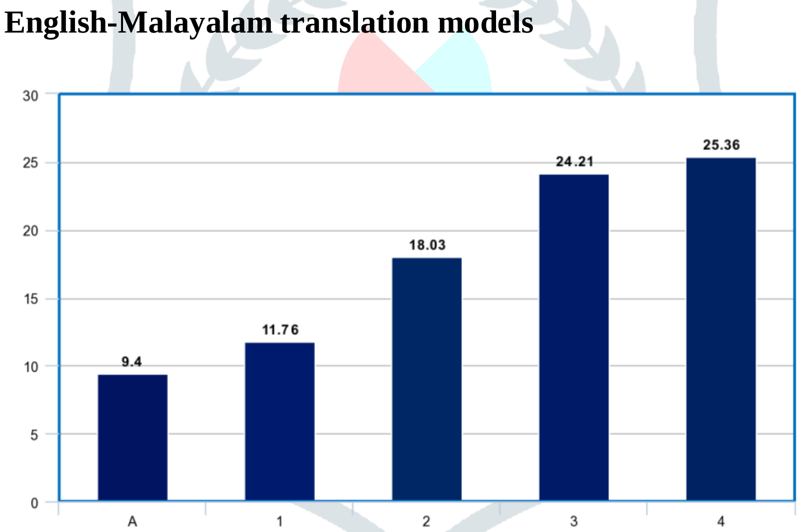

Figure 5: English-Malayalam model comparison with Google Translator Table 2

4.6 Analysis

We conducted a survey with ten random sentences from our test data and accumulated the reviews of native Tamil speaking peoples. On comparing the reviews of Google translator and our translator, it was found, that our transla-tion results were better in 60% cases than the Google trans-lator. The visualization of an Attention can be seen in 6 ofone of the sample sentences from our test data.

5 Conclusion

In this paper, we applied Neural Machine Translation (NMT) on two of the most difficult Indian language pairs (English-Tamil, English-Malayalam). We addressed the issues of data pre-processing and tokenization. To handle morphology and word complexities of Indian languages we applied pretrained fast text BPEmb, MultiBPEmb embed-dings along with multi-head self-attention which outperformed Google translator with a margin of 3.96 and 15.96 BLEU points respectively. The same approach can be plied to other Indian languages as well. Since the accuracyof our model was fairly good, it can be used for creating English-Malayalam and English-Tamil translation applications that will be very useful in domains like tourism, education and corporate. In the future, we can also explore the possibility to improve the translation results for code- switched languages using MultiBPE and other variations.

ID	Model	BLEU Score
A	Google Translator	5.71
1	Bi-LSTM(4-Layers)+ A+ Bahdanau Attention + WE	5.96
2	Bi-LSTM(4-Layers)+ A + Bahdanau Attention + Pre-BPE(25000)	7.87
3	Bi-LSTM(6-Layer)+ A + Multi-Head Attention+ Pre-BPE(25000)	9.07
4	Bi-LSTM(6-Layer)+ A + Multi-Head self Attention + Pre-MultiBPE(100000)	9.67

Table 2: English-Tamil model comparison with Google Translator (A=Adam, WE=Word Embeddings)

ID	Model	BLEU Score
A	Google Translator	9.40
1	Bi-LSTM(4-Layers)+ A+ Bahdanau Attention + WE	11.76
2	Bi-LSTM(4-Layers)+ A + Bahdanau Attention + Pre-BPE(25000)	18.03
3	Bi-LSTM(6-Layer)+ A + Multi-Head Attention+ Pre-BPE(25000)	24.21
4	Bi-LSTM(6-Layer)+ A + Multi-Head self Attention + Pre-MultiBPE(100000)	25.36

Table 3: BLEU Score of English-Malayalam translated system. (A=Adam, B= Bahdanau, WE=Word Embedding)

Attention Visualization

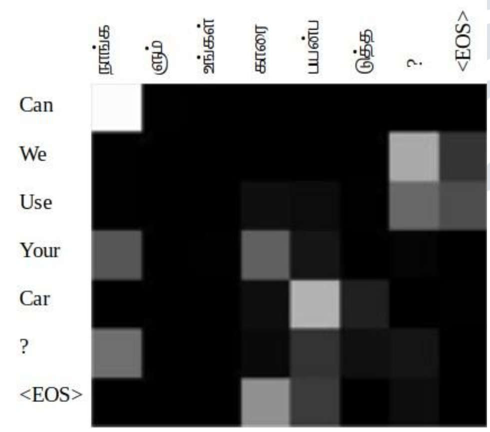

Figure 6: Attention visualization of English-Tamil sentencepair from our test data

6 Biblographical References

[1] Artetxe, M., Labaka, G., Agirre, E., and Cho, K. (2017). Unsupervised neural machine translation.*CoRR*, abs/1710.11041.

[2] Booth, A. D. (1955). Machine translation of languages, fourteen essays. Cho, K., Van Merriënboer, B., Bahdanau, D., and Ben- gio, Y. (2014). On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprintarXiv:1409.1259*.

[3] Fadaee, M., Bisazza, A., and Monz, C. (2017). Data aug- mentation for low-resource neural machine translation. In *Proceedings of the 55th Annual Meeting of the Asso- ciation for Computational Linguistics (Volume 2: Short Papers)*, pages 567–573, Vancouver, Canada, July. Asso-ciation for Computational Linguistics.

[4] Gage, P. (1994). A new algorithm for data compression. *The C Users Journal*, 12(2):2338.

[5] Ghosh, S., Thamke, S., et al. (2014). Translation of telugu-marathi and vice-versa using rule based machine

transla-tion. *arXiv preprint arXiv:1406.3969*.

[6] Hans, K. and Milton, R. (2016). Improving the performance of neural machine translation involv- ing morphologically rich languages. *arXiv preprint arXiv:1612.02482*.

[7] Heinzerling, B. and Strube, M. (2018). BPEmb: Tokenization-free Pre-trained Subword Embeddings in 275 Languages. In Nicoletta Calzolari (Conference chair), et al., editors, *Proceedings of the Eleventh Inter-national Conference on Language Resources and Eval- uation (LREC 2018)*, Miyazaki, Japan, May 7-12, 2018. European Language Resources Association (ELRA).

[8] Jawaid, B. and Zeman, D. (2011). Word-order issues in English-to-Urdu statistical machine translation.

[9] Khan, N. J., Anwar, W., and Durrani, N. (2017). Machine translation approaches and survey for indian languages. *arXiv preprint arXiv:1701.04290*.

[10] Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311318. Association for Computational Linguistics.

[11] Patel, R. N., Pimpale, P. B., et al. (2017). Mtil17: English to indian langauge statistical machine translation. *arXiv preprint arXiv:1708.07950*.

[12] Patel, R., Pimpale, P., and Mukundan, S. (2018). Machine translation in indian languages: Challenges and resolu- tion. *Journal of Intelligent Systems*, 09.

[13] Pathak, A. and Pakray, P.). Neural machine translation forindian languages. *Journal of Intelligent Systems*.

[14] Raju, B. N. V. N. and Raju, M. S. V. S. B. (2016). Statistical machine translation system for indian languages. In *2016 IEEE 6th International Conference on AdvancedComputing (IACC)*, pages 174–177, Feb.

[15] Ramasamy, L., Bojar, O., and Zˇ abokrtsky´, Z. (2012). Mor- phological processingfor english-tamil statistical ma- chine translation. In Proceedings of the Workshop on Machine Translation and Parsing in Indian Languages (MTPIL-2012), pages 113–122.

[16] Ramesh, S. H. and Sankaranarayanan, K. P. (2018). Neu- ral machine translation for low resource languages using bilingual lexicon induced from comparable corpora. In Proceedings of the 2018 Conference of the North Amer- ican Chapter of the Association for Computational Lin- guistics: Student Research Workshop, pages 112–119.

[17] Revanuru, K., Turlapaty, K., and Rao, S. (2017a). Neural machine translation of indian languages. In Proceedings of the 10th Annual ACM India Compute Conference on ZZZ, pages 11–20. ACM.

[18] Revanuru, K., Turlapaty, K., and Rao, S. (2017b). Neural machine translation of indian languages. 11.

[19] Salunkhe, P., Kadam, A. D., Joshi, S., Patil, S., Thakore, D., and Jadhav, S. (2016). Hybrid machine translation for english to marathi: A research evaluation in machine translation:(hybrid translator). In Electrical, Electronics, and Optimization Techniques (ICEEOT), International Conference on, pages 924–931. IEEE.

[20] Sennrich, R., Haddow, B., and Birch, A. (2015). Neural machine translation of rare words with subword units. arXiv preprint arXiv:1508.07909.

[21] Shilon, R. (2011). Transfer-based Machine Translation between morphologically-rich and resource-poor lan- guages: The case of Hebrew and Arabic. Ph.D. thesis, Citeseer.

[22] Somers, H. (2003). An overview of ebmt. In Recent ad- vances in example-bas ed machine translation, pages 357. Springer.

[23] Tiedemann, J. (2012). Parallel data, tools and interfaces in opus. In Nicoletta Calzolari (Conference Chair), et al., editors, Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12), Istanbul, Turkey, may. European Language Resources Association (ELRA).

[24] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u., and Polosukhin, I. (2017). Attention is all you need. In I. Guyon, et al., editors, Advances in Neural Information Processing Sys- tems 30, pages 5998–6008. Curran Associates, Inc.

[25] Wong, F., Dong, M., and Hu, D. (2006). Machine trans- lation using constraint-based synchronous grammar. Ts- inghua Science and Technology, 11(3):295–306.

[26] ZdenekZˇ abokrtsky`, L. O. (2012). Morphological process- ing for english-tamil statistical machine translation. In 24th International Conference on Computational Lin- guistics, page 113.