IJHSR

ijhighschoolresearch.org

RESEARCH ARTICLE

Effective Approaches and Challenges in Hindi-English Neural Machine Translation

Aakrit Singhal

Podar International School, Saraswati Road Santacruz West, Mumbai, Maharashtra, 400054, India; aakritsinghal44@gmail.com

ABSTRACT: With the rise in the use of Indian languages, visualizing and improving translation tasks is essential in order to narrow the gap between the comprehension of these languages globally. In this paper, approaches of sequence-to-sequence models including attention mechanism and long short-term memory are explored and compared with other state-of-the-art models such as transformers. Previous work has explored multiple Indian languages including Bengali, Hindi, Marathi, Telugu, and Kannada. Due to the overall increase cases of use of the Hindi Language, a comparison between models is done through a custom pre-processing of a Hindi-English dataset and normalizing the data to the required needs. While some systems are focused on specific end-to-end language translations, some are large-scale systems with translation capabilities of multiple languages. The paper proposes that adding attention to the basic encoder-decoder architecture can not only improve the translation tasks with higher accuracies, but also reduce training times and make it possible to train with larger datasets. Additionally, certain challenges with Hindi translation such as word sense disambiguation are also explored by looking at examples of previous work and some of the results produced by them.

KEYWORDS: Robotics and Intelligent Machines; Machine Learning; Natural Language Processing; Machine Translation; Attention Models.

Introduction

Neural Machine Translation (MT) is one of the essential applications of Natural Language Processing that has been ongoing for the past 50 years. During this time, there have been several advancements by linguists and computer scientists in the field. The main aim of MT models is to translate literature or a given piece of text from a source language to a target language, thereby making the text available and usable for a larger group of people globally. Making rich resources of literature available to various target groups helps to remove language barriers around the world.

Since India is a large multilingual society, there is great demand for the translation of documents from one language to another language. There are 22 constitutionally approved languages,[1] which are officially used in different states with about 1652 dialects spoken by different communities and around 14 Indic scripts. All of these languages are rich in content and well-developed. However, most of these languages do have similar grammatical structures and hence similarly comprehensible scripts of literature.

In the last two decades, most of the machine translation systems were based on statistical machine translation approaches. In these systems,[2,4] the basic units of the translation process are phrases and sentences. The first neural approach started with a basic feed forward network, but translation between sentences of different lengths required the innovation of encoder-decoder models.

The first implementation of this type of model used continuous recurrent representations to build robust language models by capturing syntax, semantics, and morphology. After this, the LSTM (Long Term Short Memory) network was introduced,[5] which is a type of RNN that uses special units in addition to standard units. LSTMs include a memory cell that can maintain information in the memory for long periods of time, and a set of gates are used to control when the information enters the memory, when it outputs, and when it is forgotten. This architecture helps to learn longer-term dependencies. However, a few improvisations that were needed in RNNs then marked the beginning of Attention models.[6] The attention mechanism allows the decoder to look at an entire sentence and selectively extract the information needed during decoding. And while they may work for shorter sentences, RNN based architectures are hard to parallelize and inefficient in learning long-term dependencies within the input and output sequences. Hence, transformers use multiple attention distributions and multiple outputs for a single input in order to improve the problem.

With the availability of such models, MT has become an emerging research area in NLP for Indian languages. Indian language processing started more than a decade ago. There have been a number of attempts in MT for English to Indian languages and Indian languages to Indian languages using different approaches.[7,9] Hindi-English translation can be considered a relatively high resource pairing given the increase in the creation of parallel sentences as compared to other less common Indian Languages.[10]

Many researchers are working on the task of NMT for Indian languages lately. The Technology Development for Indian Languages (TDIL) Programme initiated by the Ministry of Electronics & Information Technology, Govt. of India is a prime example of state-of-the-art machine translation projects being conducted nationally.[11] TDIL has the aim of developing

DOI: 10.36838/v4i2.16

Information Processing Tools and Techniques to facilitate human-machine interaction without language barriers. It also serves in creating and accessing multilingual knowledge resources and integrating them to develop innovative user products and services.[12] Adding to TDIL, many such initiatives and experimentations have been undertaken for Indian language processing.

Previous work has made the use of the similarities and data availability of Hindi to produce efficient Hindi-English translation systems. The ANGLA-BHARTI represents a machine-aided translation methodology specifically designed for translating English to Indian languages.[13] It uses a pattern directed approach using context free grammar-like structures. It analyzes English only once and creates an intermediate structure called PLIL (Pseudo Lingua for Indian Languages). The PLIL structure is then converted to each Indian language through a process of text-generations. Additionally, machine translation of Bilingual Hindi-English text, that is the mixing of both languages called Hinglish is also proposed in previous work apart from singular English-Hindi NMT models.[14] It aims to present a mechanism for machine translation of Hinglish to pure (standard) Hindi and pure English forms.

Another line of work explores other traditional languages such as Bengali for aiding in specific Indian translation tasks, both between two Indian languages, and from English to an Indian language. The VAASAANUBAADA is a project to translate bilingual Bengali-Assamese news texts using the Example-Based Machine Translation technique.[15] The work involves machine translation of bilingual texts at the sentence level. Moreover, the Anubharti Technology is an English to Bangla translation system that takes a paragraph of English sentences as input sentences and produces equivalent Bangla sentences.[16] EB-ANUBAD system is comprised of a preprocessor, morphological parser, semantic parser using English word ontology for context disambiguation, an electronic lexicon associated with grammatical information and a discourse processor, and also uses a lexical disambiguation analyzer.

Along with specific language models, previous work has also been aimed at larger systems for multiple Indian languages. The Anusaaraka system makes text in one Indian language accessible in another Indian language.[17] The machine presents the source text in a language close to the target language. In the image, some constructions of the source language (which do not have equivalents) spill over to the output. Anusaarakas are currently being built for various Indian Languages. The approach for the translation in this system is divided in two parts: a) The Anusaaraka system which is based on language knowledge, and b) the domain specific knowledge based on world knowledge, statistical knowledge etc.

While these are some of the relevant works published on Indian language translation, machine translation among Indian languages is still an active research area due to the semantic and word-based ambiguities involved with a lot of the languages. Certain works talk about resource constrained word sense disambiguation (WSD), which is related to lexical/sense ambiguity in languages.[18] Moreover, it also explains the stages of NLP and ambiguities such as morphology analysis of speech tagging, named entity recognition, parsing (deep and shallow), semantics extractions, pragmatics, and discourse processing.

In addition, English to Indian language machine translation poses the challenge of structural and morphological divergence.[19] Structural difference is due to the dissimilar word orders in these languages. Specifically, English uses the subject-verb-object order, whereas most of the Indian languages primarily use subject-object-verb. The suffix separation is used to tackle the morphological divergence between English and highly agglutinative Indian languages.

Hence it can be said that NLP in Indian languages have a long way to go before ambiguities and semantic differences can be matched for better translation accuracy. However, Indian languages do have certain datasets already, having parallel corpus of the input and output language, which makes it relatively easy to understand translation tasks currently.[20,22]

Additionally, while performing translation tasks such as part of speech tagging, parsing, etc., WSD takes a sense ID and marks it on a target word WT, where the sense ID comes from Wordnet. Below is an example:

Example 1: Conversation between a mother and her son:

Mother: Son, get up quickly. The school is open today. Should you bunk? Father will be angry. Father John complained to your father yesterday. Aren't you afraid of the principal?

Son: Mummy, it's a holiday today!

In example 1 as shown,[18] the word 'Son' has a sense of ambiguity, whereas in WordNet, it has two senses – one of an offspring and the other of the Son of God. Hence, as clearly seen, the first lingual sense should be applied here. Next comes the phrasal verb/multi word which is 'get up'. We cannot use the individual meanings of 'get' and 'up' since the phrase is supposed to be considered as a single entity to understand its true meaning. Individually extracting the meaning of both words would not make factual sense to the phrase. Hence, such ambiguities of grammatical sentence structure and multiple semantics of words make it a challenging translation task.

With so many languages at play, Hindi-English translation serves as the benchmark to connect Indian languages with a high resource language like English and develop the translation task to reach better performance and accuracy with seq2seq models. With the rising concerns of translating between diverse Indian Languages, there are fewer approaches that discuss the approaches and challenges in the task, especially for Hindi and English translation. As there is a rising need to explain the aforementioned problems in a concise and effective manner, this paper aims to compare effective approaches and the challenges of MT between Indian languages due to the lack of availability of parallel data and precise comparison of various language models and architectures in the open source today. This is done through comparing recurrent neural networks with attention encoder-decoder models.

Methods

Dataset:

The experiments in this paper use the IIT Bombay English-Hindi Parallel Corpus.[23] This dataset consists of an

English-Hindi Parallel Corpus that has been pre-processed for machine translation. The Hindi side of the training, dev, test sets as well as the monolingual corpus have been normalized to ensure canonical Unicode representation using the Indic NLP Library.[24] The corpus contains 1.49 million parallel segments. The existing sources used to compile the corpus were: (a) Judicial domain corpus 1 - translation of legal judgements by in-house expert translators, (b) Judicial domain corpus 2 - student translated sentenced, (c) Mahashabdkosh - an online official terminology dictionary consisting of English and Hindi terms,[25] (d) Indian government corpora, (e) Hindi-English Linked WordNet - contains bilingual dictionary entries from hind and English linked wordnets,[26] and (f) Gyan-Nidhi Corpus - multilingual parallel corpus of Hindi and English sentences that combines sentence-length models and word-correspondence based models, and requires no language or corpus specific knowledge. The moses tokenizer for English, and the IndicNLP tokenizer for Hindi was used to prepare the data.[27,28] Table 1 shows the size of the dataset used for both languages.

Table 1: Language specific statistics of dataset.[23]

	Languages	Train	Test	Dev
Sentences		1492827	2507	520
Tokens	English	20667259	57803	10656
	Hindi	22171543	63853	10174
Types	English	250782	8957	2569
	Hindi	343601	8489	2625

Experiments:

In order to perform the translation task, two approaches have been used: Recurrent neural networks (RNN) such as Long Short-Term memory (LSTM) and Attention Models. Varied results and challenges have been observed with both of these methods. To understand machine translation models better, a sequence to sequence (seq2seq) network is one in which two RNNs work together to translate one sequence to another.[29] Here, an encoder network maps the input sequence to a vector of a fixed dimensionality and the decoder network decodes the target sequence from the vector.

In order to train the LSTM model, the preprocessing of data was done by compiling pairs of English and Hindi sentences and shuffling them. Below are two sample pairs created wherein the English sentence is the input, and the translated target sentence is in Hindi.

[('he disliked that old black automobile', 'उन्होंने उस पुराने काले ऑटोमोबाइल को नापसंद किया।'), ('they dislike peaches pears and apples', 'वे आड़ू, नाशपाती और सेब को नापसंद करते हैं।'),...]

For target sentences, a <START> token and <END> token will be added for the sequence to know when to start and stop the text generation. Two different approaches have been taken to process the data. One of them is character level predictions. Here, the punctuation is separated from the words.

Hence, 'शरद ऋतु के दौरान एकजुट राज्य कभी-कभी अद्भुत होते हैं, लेकिन यह आमतौर पर जून में गोला होता है।' becomes 'शरद ऋतु के दौरान एकजुट राज्य कभी - कभी अद्भुत होते हैं , लेकिन यह आमतौर पर जून में गोला होता है ।' .

On the other hand, in the world level predictions the target sequence is appended without change, and this is commonly used to predict a fixed set of words from the training corpus. With both the input and target tokens, feature dictionaries are then created for both the tokens that store key-pair values. This helps encode the input sentences into one-hot vectors.

Coming to the encoder-decoder structure, the encoder requires an input layer that holds the one-hot vector defined by a matrix and an LSTM layer that holds hidden states. The decoder is similar to the encoder but with its own inputs.

For the attention model, the training dataset used was reduced to 124,999 sentences each for Hindi and English, while the testing dataset was unchanged and had 2507 sentences each. The original dataset for Hindi and English was merged into 2 columns in 1 file with the respective lengths. In order to train the vast amounts of sentences, the length constraint was of 22 words including punctuation and certain English prefixes such as "I am", "He is", "she is", etc. were added to filter sentences translating to these beginnings. After normalizing and filtering the text, the dataset was trimmed to 120,396 sentences which included 7732 English words and 9329 Hindi words.

The training process of the attention model used the optimizer Stochastic Gradient Descent (SGD) for the encoder and decoder, along with a learning rate of 0.01. The loss function used was Negative Log-Likelihood (NLL), along with a dropout of 0.1. Unlike a single RNN wherein the input is matched to a respective output, seq2seq models remove the barrier of sequence length and order of words, making it more accurate for translation tasks. For example:

मैं कोई अजनबी नहीं हूं - I am not a stranger

Here, most of the output words have a direct match with the input words, however the order of sentence construction is different. For instance, "am not" is before "stranger" in English, however after "अजनबी" (stranger) in the Hindi sentence. Hence, sequence to sequence models make translation easier by accounting the sentence order and not outputting single words for an input in the same order.

For every input, the encoder outputs a vector and a hidden state which is used for the next input word. The decoder takes the encoder's output vector and outputs a sequence of words to create the translation. The last output of an encoder is called a context vector since it encodes context from the entire sequence. A basic decoder in the seq2seq model uses the context vector as the initial hidden state. Hence at every step of decoding, an input token i.e., start-of-string <SOS>, and the first hidden state which is the context vector is given.

The problem with such an architecture is that the single context vector is responsible for encoding the entire sentence. Hence, an attention decoder is added to the existing model. The attention mechanism, shown in Figure 1, allows that with each output of the decoder, it can focus on specific parts of the encoder's output at every step. This helps lower the burden on the decoder for each step. In order to achieve this, special

attention weights are calculated, which are then multiplied by the encoder's vectors to get weighted combinations.

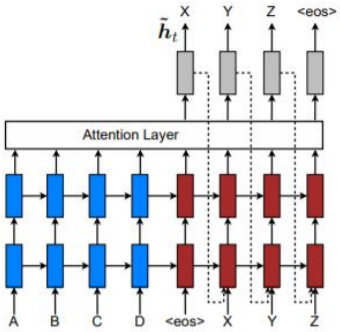

Table 1: Encoder-decoder with attention: feeding.[30]

Both the LSTM and Attention model also use the concept of teacher forcing which is used by seq2seq models during training. The teacher forcing for both was set to 0.5. The main aim of this is to use the actual target outputs as the next input instead of using the generated prediction from the decoder as the next input. This helps it to converge faster than normal.

Results and Discussion

Prediction Analysis:

By collecting the data for both the LSTM and Attention model experiments, getting training and validation accuracies and losses made it clear that the attention mechanism was generally more successful in predicting the correctly translated word. As the metrics and data processing done was not similar to the original paper, the results with that could not be fully compared. However, for the attention model, the cumulative n-gram scores and the BLEU score were calculated for sentences, and below is the data for two of those:

Table 2: Prediction Scores of 2 sentences. This shows the respective cumulative n-gram and the BLEU scores of both sentences to compare the efficacy and accuracy of translation from Hindi to English.

	Prediction 1	Prediction 2
English	existing message pane	smooth images when zoomed - in
Actual	मौजूदा संदेश पट्टी	बड़ा किये जाने पर मृदु बिंब (_ i)
Predicted	संदेश संदेश पट्टी	बड़ा किये जाने पर मृदु है (_ i)
Cumulative 1-gram	0.6667	0.8750
Cumulative 2-gram	0.5774	0.7906
Cumulative 3-gram	0.6959	0.6812
Cumulative 4-gram	0.7598	0.5946
BLEU Score	0.7598	0.5946

In the first example in Table 2, the translation of the first word i.e., "existing" is missing, while in the second example the last word is missing. However, given the small dataset the model was trained on, the n-gram scores and translation task were carried out with a high enough accuracy.

The attention mechanism can also be individualized. Since it is used to weight specific encoder outputs of the input sentence, it is helpful to look at the input token the network is focused on from each time step.

Figure 2 is generated through the translation of "did you do your". This shows the focus through the similarities between the predicted output and the target sentence at each axis. Hence, the diagonal focus of the visualization helps to understand the mapping of each word of the prediction with the actual output. In this case, the starting portions of the sentence were more accurate, as can be seen by the overall diagonal color grading, representing the input words matching the translated sentences at each time step.

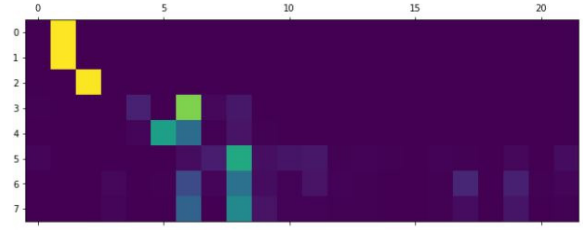

Figure 2: Visualizing Attention of a sentence. This shows the level of translation at each step in a sentence, with the color grading indicated successful translations.

Moreover, the understanding can be enhanced by adding axes to the translation sentences, which further improves comparisons by showing word by word analysis of input and output sentences.

Example 1:

input = at least two file names are equal

output = कम से कम फाइल नाम

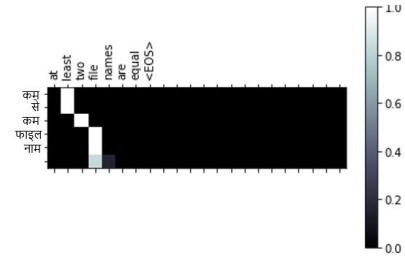

Figure 3: Example 1 sentence. The white bands show the level to which the sentence matches. Here, the translation shows some problems with a few words, while it is accurate for the majority of the sentence.

The white areas in Figure 3 represent accurate translation as shown by the grading key on the right. Upon viewing the translation, the words "are equal" and "two" are not translated by the model, leaving "at least file names" as the translated output, hence the last 2 words are black in the diagram.

Example 2:

input = weather: snow

output = मौसमः बर्षा

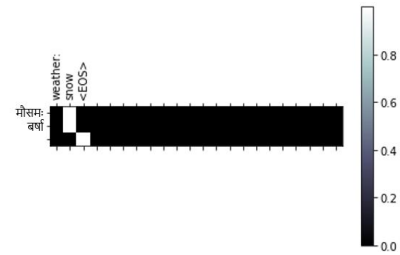

Figure 4: Example 2 sentence. This sentence shows the accurate translation of both words involved.

Both words are accurately translated in Figure 4 and thus the white marker on the complete sentence is seen.

As evident through the experimentation and previous works, attention models outperform sequence-to-sequence models. Although due to computational constraints an ideal compar

DOI: 10.36838/v4i2.16

ison between both couldn't be made, past research does show that this is the case.

Sentence Analysis:

In addition, a qualitative look at sentences produced through this attention model help to represent the overall efficacy of the model in the translation task:

Table 3: Individual Sentence BLEU Scores when compared between input and translated sentences in the attention model.

English Sentence	Predicted Hindi	Actual Hindi	BLEU Score
only one option can be given at a time	केवल एक एक में एक समय में दिया जा सकता है	केवल एक विकल्प एक समय में दिया जा सकता है	0.5878
at least two file names are equal	कम से कम फ़ाइल नाम नाम	कम से कम दो फाइल नाम समान हैं	0.6664
existing message pane	संदेश संदेश पट्टी	मौजूदा संदेश पट्टी	0.7598
pacific / apia	प्रशांत/एपिया	प्रशांत/एपिया	1.0000
smooth images when zoomed - _ in	बड़ा किये जाने पर मृदु है (_ i)	बड़ा किये जाने पर मृदु बिंब (_ i)	0.5946
work week view	कार्य कार्य दृश्य	कार्य साप्ताहिक दृश्य	0.6223

In Table 3, most sentences are predicted close to the actual one, however a common trend is the missing of one to two words, having wrong punctuation, or repeating words. For instance, in the first row, the word 'option' is missing and 'one' is repeated an extra time in Hindi. In the second row, the word 'two' is missing, and so on for all examples.

Previous works have also compared seq-2-seq models with attention and transformer models. As the experiments suggest, self-attention transformers and attention models perform better with higher BLEU scores than stand sequence models.

Table 4: BLEU scores of translation tasks of three models.[31]

Model	English-Hindi	Hindi-English
Sequence-2-Sequence	9.40	8.38
Attention Encoder-Decoder	11.59	10.13
Self-Attention Transformer	13.96	13.47

As seen in Table 4, the BLEU scores of the attention model are 21-23% more than seq-2-seq models, and the self-attention transformers are 50-60% more, making them far more accurate in both English to Hindi and Hindi to English translation tasks. Thereby, both original results and previous works highlight that adding attention to the model increases accuracy.

Conclusion

In this paper, initially the importance of neural machine translation in Indian languages was discussed. Then, it was followed by current Indian language processing models and systems that have been released for translation tasks between several other Indian languages, finally coming down to the relevance of English to Hindi translation. Then, the dataset and the experimentation including the comparison of LSTMs and Attention Models were examined. Later, an attention model with visualizing the attention was proposed and scores of multiple translation tasks performed. Finally, the results from a previous work were compared to see the performance difference in seq-2-seq with attention and transformer models. The project can be improved by using a larger dataset, which would help to train more types of sentences and increase the lingual diversity of the model. In addition, expanding to self-attention transformer or transformer models like mBART could aid with comparing other approaches for English-Hindi translation.

Acknowledgement

Aakrit Singhal thanks Melanie Subbiah, PhD Student at Columbia University for her continuous guidance and input throughout the project.

References

1. https://www.education.gov.in/hi/sites/upload_files/mhrd/files/upload_document/languagebr.pdf (accessed Jul 6, 2021).
2. Lavie, A.; Vogel, S.; Levin, L.; Peterson, E.; Probst, K.; Llitjós, A. F.; Reynolds, R.; Carbonell, J.; Cohen, R. Experiments with a Hindi-to-English Transfer-Based MT System under a Miserly Data Scenario. ACM Trans. Asian Lang. Inf. Process. 2003, 2 (2), 143–163.
3. Saini, S.; Sehgal, U.; Sahula, V. Relative Clause Based Text Simplification for Improved English to Hindi Translation. In 2015 International Conference on Advances in Computing, Communications and Informatics (ICACCI); IEEE, 2015; pp 1479–1484.
4. Saini, S.; Sahula, V. A Survey of Machine Translation Techniques and Systems for Indian Languages. In 2015 IEEE International Conference on Computational Intelligence & Communication Technology; IEEE, 2015; pp 676–681.
5. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. Neural Comput. 1997, 9 (8), 1735–1780.
6. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; Polosukhin, I. Attention Is All You Need. arXiv [cs.CL], 2017.
7. Nair, J.; Krishnan, K. A.; Deetha, R. An Efficient English to Hindi Machine Translation System Using Hybrid Mechanism. In 2016 International Conference on Advances in Computing, Communications and Informatics (ICACCI); IEEE, 2016; pp 2109–2113.
8. Laskar, S. R.; Dutta, A.; Pakray, P.; Bandyopadhyay, S. Neural Machine Translation: English to Hindi. In 2019 IEEE Conference on Information and Communication Technology; IEEE, 2019; pp 1–6.
9. Godase, A.; Govilkar, S. Machine Translation Development for Indian Languages and Its Approaches. Int. j. nat. lang. comput. 2015, 4 (2), 55–74.
10. Siripragada, S.; Philip, J.; Namboodiri, V. P.; Jawahar, C. V. A Multilingual Parallel Corpora Collection Effort for Indian Languages. In Proceedings of the 12th Language Resources and Evaluation Conference; European Language Resources Association, 2020; pp 3743–3751.
11. Jha, G. N. The TDIL Program and the Indian Langauge Corpora Intitiative (ILCI). In Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10); European Language Resources Association (ELRA): Valletta, Malta, 2010.
12. https://tdil.meity.gov.in/pdf/Vishwabharat/44/vishwabharat44.pdf (accessed Jul 6, 2021).
13. Sinha, R. M. K.; Sivaraman, K.; Agrawal, A.; Jain, R.; Srivastava, R.; Jain, A. ANGLABHARTI: A Multilingual Machine Aided Translation Project on Translation from English to Indian Languages. In 1995 IEEE International Conference on Systems, Man and Cybernetics. Intelligent Systems for the 21st Century; IEEE, 2002; Vol. 2, pp 1609–1614 vol.2.

14.Sinha, R. Machine Translation of Bi-Lingual Hindi-English (Hinglish) Text. 2005.
15.Vijayanand, K.; Choudhury, S. I.; Ratna, P. VAASAANUBAADA: Automatic Machine Translation of Bilingual Bengali-Assamese News Texts. In Language Engineering Conference, 2002. Proceedings; IEEE Comput. Soc, 2003; pp 183–188.
16.Saha, G. The EB-ANUBAD Translator: A Hybrid Scheme. 2005. https://doi.org/10.1007/BF02842224.
17.Bharati, A.; Chaitanya, V.; Kulkarni, A. P.; Sangal, R.; Rao, G. U. Anusaaraka: Overcoming the Language Barrier in India. arXiv [cs. CL], 2003.
18.Bhattacharyya, P. Natural language processing: A perspective from computation in presence of ambiguity, resource constraint and multilinguality https://www.cse.iitb.ac.in/~pb/papers/csi-nlp-pb-8aug12.pdf (accessed Jul 6, 2021).
19.Patel, R. N.; Pimpale, P. B.; Sasikumar, M. Machine Translation in Indian Languages: Challenges and Resolution. J. Intell. Syst. 2019, 28 (3), 437–445.
20.Deep, K.; Kumar, A.; Goyal, V. Development of Punjabi-English (PunEng) Parallel Corpus for Machine Translation System. Int. j. eng. technol. 2018, 7 (2), 690.
21.Goyal, V.; Sharma, D. M. The IIIT-H Gujarati-English Machine Translation System for WMT19. In Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1); Association for Computational Linguistics: Stroudsburg, PA, USA, 2019
22.Jadhav, S. A. Marathi to English Neural Machine Translation with near Perfect Corpus and Transformers. arXiv [cs.CL], 2020.
23.Kunchukuttan, A.; Mehta, P.; Bhattacharyya, P. The IIT Bombay English-Hindi Parallel Corpus. In Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018); European Language Resources Association (ELRA).
24.Kunchukuttan, A. Indic_nlp_library.
25.Dictionary http://e-mahashabdkosh.rb-aai.in/ (accessed Jul 6, 2021).
26.Kanojia, D.; Patel, K.; Bhattacharyya, P. Indian Language Wordnets and their Linkages with Princeton WordNet https://www.aclweb.org/anthology/L18-1728.pdf (accessed Jul 6, 2021).
27.Koehn, P.; Hoang, H.; Birch, A.; Callison-Burch, C.; Federico, M.; Bertoldi, N.; Cowan, B.; Shen, W.; Moran, C.; Zens, R.; Dyer, C.; Bojar, O.; Constantin, A.; Herbst, E. 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; Association for Computational Linguistics: Prague, Czech Republic, 2007; pp 177–180.
28.Kakwani, D.; Kunchukuttan, A.; Golla, S.; Gokul; Bhattacharyya, A.; Khapra, M. M.; Kumar, P. IndicNLPSuite: Monolingual Corpora, Evaluation Benchmarks and Pre-Trained Multilingual Language Models for Indian Languages. In Findings of the Association for Computational Linguistics: EMNLP 2020; Association for Computational Linguistics: Stroudsburg, PA, USA, 2020.
29.Sutskever, I.; Vinyals, O.; Le, Q. V. Sequence to sequence learning with neural networks https://papers.nips.cc/paper/2014/file/a14ac55a4f27472c5d894ec1c3c743d2-Paper.pdf (accessed Jul 6, 2021).
30.Luong, T.; Pham, H.; Manning, C. D. Effective Approaches to Attention-Based Neural Machine Translation. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing; Association for Computational Linguistics: Stroudsburg, PA, USA, 2015; pp 1412–1421.
31.Srivastava, S.; Tiwari, R. Self-Attention Based End-to-End Hindi-English Neural Machine Translation. arXiv [cs.CL], 2019.

Author

Aakrit Singhal is a senior at Podar International School, Mum bai, India. He is passionate about the field of computer science, with research experience in fields of NLP and Computer Vision at sever al research institutes. Last year, he led and won 2 international hack athons on COVID-19 held by the Massachusetts Institute of Tech nology. Aakrit is also the founder of STEM Elevate, a non-profit for education of underprivilege children in Mumbai. In the future, he hopes to pursue higher studies in the fields of machine learning and its applications.

DOI: 10.36838/v4i2.16