ISSN: 0974-5823 Vol. 6 No. 3 December, 2021

International Journal of Mechanical Engineering

Paraphrasing in Hindi Using Attention Model

Manav Jain[1, a)], Ashay Mehta[1, b)], Simrn Gupta[1, c)] and Pankaj Sonawane[1, d)]

Author Affiliations

[1]*Computer Engineering, D.J Sanghvi College of Engineering, Mumbai, India*

Abstract. Natural language processing (NLP) has been applied across various linguistic and semantic applications covering machine translation, question- answering and paraphrasing among others. The job of automatically extracting or spawning lexical equivalences for the different word constituents, expressions and sentences in the language is a crucial segment of the natural language processing, which is being progressively used to accentuate the performance of a plethora of NLP applications like data augmentation, text summarization, etc. Developing such systems for high-resources languages has been dominant, with a need to focus on low-resource languages. This paper revolves around building a paraphrasing model for Hindi, leveraging recurrent neural networks of the Long Short Term Memory and the Gated Recurrent Unit with Adaptive Attention. This task is complicated by sentence structuring and word relocation, which the paper means to counter. Evaluation has been carried out using BLEU and METEOR scores, both the models performed well. The better model resulted being LSTM with applied attention.

INTRODUCTION

Textual expressions which use different surface forms but possess the same semantic meaning are called paraphrases. [1] Paraphrasing has grown into a competent application in the field of NLP, as it continues to intrigue researchers, seasoned and aspiring. It is viewed as a potential application to diverse problems in the domain, including but not limited to, machine translation, semantic parsing, question-answering and summarization. [2] Most of the research has been focussed on the rephrasing of high-resource languages, with relatively limited review for low-resource languages. One contributing factor to this impediment is the lack of available corpuses in low-resource languages. A compelling motive for systematic study in paraphrasing of low-resource languages is the positive impact it may have on the segment of society which finds high-resource languages abstruse. This paper presents a paraphrase model for Hindi, which is widely understood in India, with over 500 million native speakers.

Several studies on neural machine translation and paraphrasing have been employed using encoder-decoder models. The encoder accepts the source sentence as input, which is encoded into a fixed-length vector. The decoder neural network then decodes the encoded vector to produce a translation as output. However, an issue with this approach occurs as there is a need to produce a fixed-length vector with all necessary information, irrespective of length of the sentence. This is especially a problem when there are sentences longer than those in the training corpus. Therefore, this paper applies the attention approach proposed by Bahdanau [3], which does not attempt to produce a single fixed-length vector, rather the words are predicted based on context vectors attributed to a set of positions with germane information, and all words which were generated previously.

Encoder-decoder models vastly leverage Recurrent Neural Networks (RNNs) which can capture distant range dependencies among subsequent observations. Nonetheless, they encounter the complications of "vanishing and exploding gradient" problems, which act as a deterrent for the neural network to capture long-range dependencies [4]. This study makes use of RNNs with the "gating" approach, namely the Long Short Term Memory Unit (LSTM) and the Gated Recurrent Unit (GRU), which were built to offset these limitations.

Paraphrasing in Hindi can become involved due to the issues. First, in sentence organization, in Hindi, the words are organized in subject-object-verb format, whereas in English, the words are formatted in subject-verb-

object format. For example, "मैंरोज दूधपीता हूं।"and "I drink milk every day." Secondly, prepositions are replaced by specifying the postposition. For instance, "किताब मेजिै<u>ऊपर</u> हैं।"and "The book is <u>on</u> the table." Finally, the words can be relocated without changing the meaning of the sentence. For example, "कमठाइयाूंमुझेपसूंदहैं।"can be changed to "मुझेकमठाइयाूंपसूंदहैं।".

For assessment, the paper utilizes the widely known automatic evaluation metrics in the natural language processing domain: METEOR (Metric for Evaluation of Translation with Explicit ORdering) and BLEU (Bi-Lingual Evaluation Understudy). It has been shown by the research work carried out earlier that the above metrics show promising results for the paraphrase recognition task and also correlates well with human interpretations in ascertaining the paraphrases generated.

RELATED WORK

The paraphrasing problem for English has been solved using the statistical machine translation approach by QUIRK, C. et al., [7] in the past. This method uses a sentence decoder which is monotonic, to generate output sentences having the same meaning as the input sentence. The problem is also solved by applying reinforcement learning by Jiang L. et al. [10], which uses the sequence-to-sequence model as the generator and paraphrase identifier as the evaluator.

Many studies that have incorporated the Neural Machine Translation(NMT) approach have provided excellent results. Like, Datla V. [5] et al. used an attention-based bidirectional recurrent neural network. In addition, the system used a character-level as well as a word-level encoder-decoder framework. Also, the approach proposed by Aaditya Prakash, S., et al., [6] used a multilayer LSTM model with a residual connection between the layers for faster training.

Machine Translation also inspired the other approaches. For Example, Elozino Egonmwan, Y. et al., [8] merged to NMT models sequence to sequence and Transformer. The system consisted of 6 multi-head self-attention layers followed by a Gated Recurrent Unit Layer. Another approach presented by Alex Sokolov D. et al. [12] contained a biLSTM and an LSTM layer. And, the decoder consisted of two LSTM layers. The embeddings, as well as previously generated sentences, were passed to the model. The input of the models included vectors generated by GloVe, a state of the sentence embedding model. Also, a study by Brian Thompson, M. et al. [9] used a multilingual model. They provided a simple algorithm which discouraged the production of n-grams present in the input.

Prior work on paraphrasing in Hindi by Sethi Ni et al. [11] is done using the replacement of synonyms, antonyms, and reordering the words in the sentence. However, this model is entirely dependent on the size of the database. Therefore, we choose a deep learning-based solution by applying adaptive attention to a Recurrent Neural Network.

METHODOLOGY

Dataset

The dataset used in the paper is MSCOCO (Microsoft Common Objects in Context) which is translated from English to Hindi using the Google Translate API by "https://github.com/nayeem8527/Chitra-VarNan". This paper assumes that the translations done by Google Translate API are correct and are suitable for ground truth for training our model for paraphrasing. The MSCOCO dataset is previously used to evaluate paraphrase generation methods in multiple papers for paraphrasing in English. This dataset is a standard benchmark dataset for image caption generation tasks. The dataset is human annotated for each image there are five different human interpreters. In the majority of the cases, interpreters describe the most prominent object/action in an image. Therefore, generating five captions for each image. This makes the dataset suitable for the paraphrase generation task. The

dataset contains over 82K samples for training and contains over 21K samples for validation. From the five captions accompanying each sample, we omit the fifth caption and use the first four as training instances by pairing the first caption with the second one and third caption with the fourth one.

Proposed Model

Adaptive Attention

Regular encoder-decoder approaches towards paraphrasing and neural machine translation aim to extract all the relevant information from the source sentence into a fixed-length vector, irrespective of the length of the source sentence. However, this results in difficulty for the network to address long sentences, specifically those which are longer than sentences in the corpus the model is trained on. To offset this, the adaptive attention mechanism encodes the source sentence into a sequence of vectors; subsets of these vectors are chosen adaptatively while decoding takes place.[3]

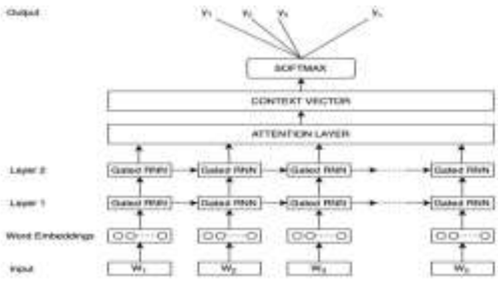

FIGURE 1. Proposed Model

Long Short Term memory (LSTM)

Recurrent neural networks suffer from the problems of "Exploding" and "Vanishing" gradients, which makes them inept to learn dependencies over long ranges. LSTMs counter this problem, maintaining a constant error flow which can be backpropagated through time and layers. They comprise three gates: the input, output and forget gates, which help to decide whether the existing memory is to be retained [4]. By passing the essential information over a long distance in an arbitrary amount of time, it captures potential long-distance dependencies. [13]

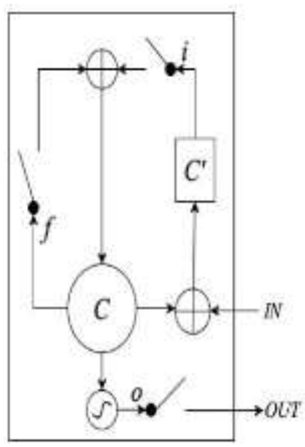

FIGURE 2. Block Diagram of LSTM Unit

Gated Recurrent Unit (GRU)

Similar to the LSTM units, GRUs address the limitations of the conventional RNN, where each recurrent unit captures dependencies adaptatively over long distance and time [13]. In this, the input and forget gates combine to form the “update” gate, with the addition of a “reset” gate [4]. The GRU is slightly different from the LSTM, in the sense that the former exposes the entire memory to the network and is simpler than the standard LSTM model.

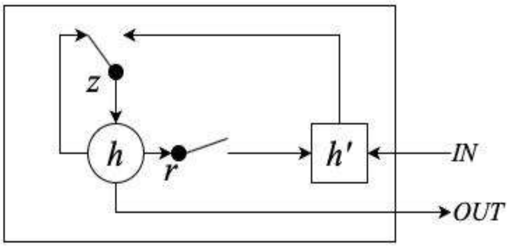

FIGURE 2. Block Diagram of GRU

This paper employs two models, Attention-based LSTM and Attention-based GRU for paraphrasing in Hindi. The LSTM and GRU models are configured for 1024 units and batch size of 64. (table)

TABLE 1. Model Hyperparameters

Parameters	LSTM (Attention)		GRU (Attention)	
Encoder Layers	1	Embedding, 1 LSTM	1	Embedding, 1 GRU
Decoder Layers	1	Embedding, 1 LSTM, 1 Dense, 1 Attention	1	Embedding, 1 GRU, 1 Dense, 1 Attention
Activation	Softmax		Softmax	
Optimizer	Adam		Adam	
Loss	Sparse Categorical Cross-Entropy		Sparse Categorical Cross-Entropy	
Batch size	64		64	
Epochs	15		15	

Attention is then applied to the above LSTM/GRU model, with the attention layer of 10 units. The alignment scores are calculated using the hidden state produced by the decoder in the previous time step and the encoder outputs, after being passed through the tanh activation function. The attention weights are obtained by softmaxing the alignment scores produced previously. Subsequently, these weights are multiplied by the encoder outputs to compute the context vector, which is inputted to the decoder with the previous decoder hidden state, resulting in a new output and so forth. [14]

Implementation Environment

The experiments have been performed in the Python environment on Google Collaboratory that provides the following configuration:

- Intel® Xeon® CPU @ 2.00GHz
- 13 GB RAM
- Tesla T4 GPU with 16 GB GDDR6 memory.

Evaluation methods

BLEU is a measurement of the equivalence between differences an automatic translation by the system and one or more human-created reference translations of the same source sentence. It considers word to word match between reference paraphrases and system generated paraphrase using the concept of modified n-gram precision and brevity penalty. The METEOR automatic evaluation metric scores machine translation hypotheses just like BLEU

by aligning them to one or more reference translations. Alignments are based on exact, stem, synonym, and paraphrase matches (using WordNet) between words and phrases. Our test set contained over 21K examples each containing five sentences each. Out of the five sentences, one sentence was used for prediction and the other four were used as hypothesis for calculation of BLEU and METEOR score. We used the “Natural Language Toolkit (NLTK)” library for calculation of the scores.

I. RESULTS AND DISCUSSION

We performed the tests mentioned above on the test data. It was visible that the model was able to generate grammatically and logically correct sentences for almost all the examples in the dataset. The models were able to replace words with their synonyms (No 1, 2 and 3 in the table). The models were also able to replace multiple words with one word (No 1 and 3 in the table). For Example, “एि व्यक्ति स्के टबोर्डचलाता है”was replaced to “स्के टबोर्डर”.

The models were also able to convert the sentence from active voice to passive voice (No 4 in the table). The models were also able to replace verbs (No 1 in the table). For example, “बॉल मारने”was replaced with “क्तवूंगलेता”.

TABLE 2. Paraphrase Example of the model

No	Source	Attention + LSTM	Attention + GRU
1	[illegible]	[illegible]	[illegible]
2	[illegible]	[illegible]	[illegible]
3	[illegible]	[illegible]	[illegible]
4	[illegible]	[illegible]	[illegible]

After evaluation of the scores, it was visible that the Attention + LSTM had a slightly better Bleu and METEOR score than the Attention + GRU model. But the predictions made by both the models were not distinguishable to humans.

TABLE 3. Metrics Comparison of LSTM and GRU models

Metric	Attention + LSTM	Attention + GRU
Bleu Score(1gram)	55.45	52.81
Bleu Score(2gram)	40.22	38.02
Bleu Score(3gram)	32.42	30.84
Bleu Score(4gram)	28.21	27.98
METEOR Score	37.68	36.37

CONCLUSIONS

This paper employs recurrent neural networks of LSTM and the GRU with Attention mechanism on the MSCOCO dataset which has been translated into Hindi. After applying the proposed model on test data, the performance was evaluated using BLEU and METEOR scores. The LSTM model with Attention applied showed the best performance. Since the models are generalized, in future they can be applied on such tasks on unavailability of other models.

ACKNOWLEDGMENT

This research did not receive any specific funding from donors in the public, commercial, or not-for-profit sectors.

REFERENCE

[1]R. Bhagat, E. Hovy, and S. Patwardhan, “Acquiring Paraphrases from Text Corpora.”

[2]J. Mallinson, R. Sennrich, and M. Lapata, “Paraphrasing Revisited with Neural Machine Translation,” Long Papers, vol. 1, pp. 881–893, 2017

[3]D. Bahdanau, K. Cho, and Y. Bengio, “NEURAL MACHINE TRANSLATION BY JOINTLY LEARNING TO ALIGN AND TRANSLATE,” 2015.

[4]R. Rana, J. Epps, R. Jurdak, X. Li, R. Goecke, and M. Brereton, “Gated Recurrent Unit (GRU) for Emotion Classification from Noisy Speech.”

[5]S. Hasan, B. Liu, J. Liu, A. Qadir, K. Lee, and V. Datla, “Neural Clinical Paraphrase Generation with Attention,” 2016.

[6]A. Prakash, S. Hasan, K. Lee, V. Datla, A. Qadir, and J. Liu, “Neural Paraphrase Generation with Stacked Residual LSTM Networks.”

[7]C. Quirk, C. Brockett, and W. Dolan, “Monolingual Machine Translation for Paraphrase Generation.”

[8]E. Egonmwan and Y. Chali, “Transformer and seq2seq model for Paraphrase Generation,” Association for Computational Linguistics, 2019.

[9]B. Thompson and M. Post, “Paraphrase Generation as Zero-Shot Multilingual Translation: Disentangling Semantic Similarity from Lexical and Syntactic Diversity.” Accessed: Sep. 14, 2021.

[10]Z. Li, X. Jiang, L. Shang, and H. Li, “Paraphrase Generation with Deep Reinforcement Learning.” Accessed: Sep. 14, 2021.

[11]N. Sethi, P. Agrawal, V. Madaan, and S. Kumar Singh, “A Novel Approach to Paraphrase Hindi Sentences using Natural Language Processing,” Indian Journal of Science and Technology, vol. 9, no. 28, Jul. 2016, doi: 10.17485/ijst/2016/v9i28/98374.

[12]A. Sokolov and D. Filimonov, “Neural Machine Translation For Paraphrase Generation.”

[13]J. Chung, C. Gulcehre, and K. Cho, “Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling.”