English to Nepali Sentence Translation Using Recurrent Neural Network with Attention

2021 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS) | 978-1-7281-8529-3/20/$31.00 ©2021 IEEE | DOI: 10.1109/ICCCIS51004.2021.9397185

Kriti Nemkul
Central Department of Computer Science and Information Technology, Tribhuvan University
Kirtipur, Nepal
Kriti.nemkul@gmail.com

Subarna Shakya
Institute of Engineering
Tribhuvan University
Pulchowk, Lalitpur
drss@ioe.edu.np

***Abstract*— Machine Translation, an automated system that intakes the text from the source language as an input, applies some computation on that input and gives the equivalent text in the target language without any human involvement. This research work focuses on developing the models for English to Nepali sentence translation incorporating Gated Recurrent Unit (GRU) and Long Short Term Memory (LSTM) with attention. Bilingual Evaluation Understudy (BLEU) Score is calculated to evaluate the efficiency of the model. Different parameters has been used to test the model. The model has been tested with neural network layer 2 and 4 and with hidden units 128, 256 and 512. The GRU cells in encoder and decoder with attention with 2 layer of neural network and 512 hidden units appears to be better in translating the English sentences into Nepali sentences with highest BLEU score 12.3.**

Keywords—Source Language; Target Language, Long Short Term Memory (LSTM), Gated Recurrent Unit (GRU), Bilingual Evaluation Understudy (BLEU)

I. INTRODUCTION

Natural Language Processing (NLP), a set of technically inspired computational techniques for automated analysis and representation of natural language enabling computers to perform a variety of tasks related to natural language, parsing, part-of-speech (POS) tagging, machine translation and dialogue systems [2]. Machine translation can be considered as an area of applied science which performs the the task of translating the input text in source language to destination language, automatically [3]. Recurrent Neural Network (RNN), the deep learning model which has been successfully used for natural language processing task such as machine translation. The encoder-decoder recurrent neural network with long short term memory (LSTM) or gated recurrent units (GRU) can be used to achieve near to state-of-art performance in machine translation task. In this procedure, neural network encoder encodes source sentence it into fix-length vector. Decoder then performs translation task by decoding the fixed-length vector into a sentence of varying length [4]. This approach of encoding the varying length whole input sentence into a fix-length vector seems to be troublesome when translating long sentences [5]. The attention based neural machine translation can be used to overcome this limitation in which rather than encoding the given sentence into a fixed – length vector, context vector is generated by the attention model which is filtered uniquely for each output time step. When a word is generated during translation, it checks for a series of positions in the source sentence where the key information is focused. The target word is then predicted by the model on the basis of the context vectors associated with those input positions and all the previously generated target words [5] [6].

II. ENGLISH TO NEPALI TRANSLATION

Machine Translation (MT) refers to an automated system that analyzes text from a Source Language (SL), applies some computation on that input and ideally generates equivalent text in a required target language (TL) without any kind of human involvement [7]. Nepali language is a mother/national language of Nepal which is Indo-Aryan language. The root of Nepali language is Sanskrit which is the classical language of India. Nepali language was previously identified as Khas Kura and written in Devanagari script [11]. The grammatical structure of the Nepali language is different than that of the English language. English word order is "subject – verb – object", but Nepali word order is "subject – object – verb".

Example,
(Source language) English sentence: I eat rice.
The source sentence after the translation,
(Target language) Nepali sentence: म भात खान्छु ।

Here, if Nepali sentence is expressed as S + V + O, then it will be म खान्छु भात ।, which will be both semantically and syntactically wrong. Similarly, for the fluent translation, original input text must be read by a translator (human or machine) and understand the situation to which it relates and find a corresponding text in a destination language that exactly or nearly describes the similar or same situation. For example, an English word "you" while referring to single person can be translated into Nepali language as either "तिमि" or "तपाई" or "तँ". Translation procedure sometimes finds it difficult to make a choice.

III. LITERATURE REVIEW

R. Agrawal and D. M. Sharma has used sequence – to – sequence models using Recurrent Neural Network for English

ISBN: 978-1-7281-8529-3/21/$31.00 ©2021 IEEE

– Hindi machine translation task. They experiment with Gated Recurrent Units (GRU), Long Short-Term Memory Units (LSTM), Bidirectional Long Short Term Memory Units (BiLSTM) and the attention mechanism. Long Short-Term Memory (LSTM) units, a type of RNNs which are very effective at retaining information over time-steps. Recurrent Neural Networks works well in English-Hindi machine translation task. The bi-directional LSTM units when complemented with the attention mechanism perform best, especially on compound sentences [1]. K. H. cho, D. Bahdanau, F. Bougares, H. Schwenk and Y. Bengio has proposed RNN Encoder – Decoder neural network architecture that is capable of mapping from a sequence of arbitrary length to another sequence, possibly from another series, of an arbitrary length with the ability to either score a pair of sequences or create a target sequences provided a source sequence [17]. D. Bahdanau, K. H. Cho and Y. Bengio had conjectured that it was problematic to use fixed-length context vector to translate long sentences. The simple encoder–decoder model is extended by enabling the model (soft-) search for a collection of input terms, or their annotations computed by an encoder, while producing the target word. This liberates the model from having to encode a whole source sentence into a fixed-length vector, and also allows the model concentrate only on the information relevant to next target word generation. The proposed model is named as RNNsearch, which outperforms the traditional encoder–decoder model (RNNencdec) on the task of English – to – French translation significantly, regardless of the sentence length [5]. O. Bojar and et al. presents HindEnCorp, Hindi - English parallel corpus, and HindMonoCorp, a monolingual Hindi corpus to train statistical machine translation system [23]. The rule based English to Nepali machine translation project "Dobhase" was developed by Madan Puraskar Pustakalaya (MPP, http://www.mpp.org.np) and in collaboration with the Kathmandu University (http://ku.edu.np) [9]. K. Nemkul and S. Shakya presented low resource English to Nepali sentence translation using LSTM with attention showing the result with BLUE score 8.9 for very small corpus size [1]. Interlingua approach of machine translation, especially the Nepali generator part was proposed by B. K. Sanat, K. Bista in which UNL (Universal Networking Language) interlingua has been used. They used techniques of syntax planning and morphology generation for a translation task [25].

IV. Data Collection

During this study, English – Nepali parallel corpus of 20000 has been collected from different sources. The data set arranged such that file contains set of English sentences and another file contains the corresponding Nepali sentences aligned parallel. The English-Nepali parallel corpus was collected from the Kathmandu University, NLP Lab research center, "English-Nepali Parallel Corpus", ELRA catalogue [29], Opus data set [30]. The collected data are merged and preprocessed to remove the dirty data.

V. Research Methodology

A. RNN Encoder – Attention – Decoder Model with Gated Recurrent Unit

1) Encoder

The bidirectional RNN consists of forward and backward RNN's. The input sequence is read by the forward RNN $\vec{f}$ in order (from x_1 to x_{T_x}) and sequence of forward hidden states ($\overrightarrow{h_1}$, …, $\overrightarrow{h_{T_x}}$) are computed. Similarly, the backward RNN $\overleftarrow{f}$ takes the sequence in the reverse order (from x_{T_x} to x_1) which calculates the backward hidden state sequence ($\overleftarrow{h_1}$, …, $\overleftarrow{h_{T_x}}$).

The 1 – of – K coded word vectors from the source sentence are taken as input by the model,

$$x = \left(x_1, \dots, x_{T_x}\right), x_i \in \mathbb{R}^{K_x} \quad (1)$$

and the model generates the translated sentence of 1 – of – K coded word vectors,

$$y = \left(y_1, \dots, y_{T_y}\right), y_i \in \mathbb{R}^{K_y}, \quad (2)$$

where K_x and K_y represents the vocabulary sizes of source and target languages, respectively. T_x and T_y denotes the length of source and target sentences, respectively.

First, the forward states of the recurrent neural network are computed as,

$$\overrightarrow{h_\iota} = \begin{cases}(1 - \overrightarrow{z_\iota}) \circ \vec{h}_{i-1} + \overrightarrow{z_\iota} \circ \underline{\overrightarrow{h_\iota}} & ,if\ i > 0 \\ 0 & ,if\ i = 0\end{cases} \quad (3)$$

where,

$$\underline{\overrightarrow{h_\iota}} = tanh(\overrightarrow{W}\overline{E}x_i + \vec{U}[\vec{r}_i \circ \vec{h}_{i-1}]) \quad (4)$$

$$\overrightarrow{z_\iota} = \sigma\,(\overrightarrow{W_z}\overline{E}x_i + \vec{U}_z\vec{h}_{i-1}) \quad (5)$$

$$\overrightarrow{r_\iota} = \sigma(\overrightarrow{W_r}\overline{E}x_i + \vec{U}_r\vec{h}_{i-1})\ . \quad (6)$$

$\overline{E} \in \mathbb{R}^{m \times K_x}$ is the word embedding matrix. $\overrightarrow{W}$, $\overrightarrow{W_z}$, $\overrightarrow{W_r}$ $\in$ $\mathbb{R}^{n \times m}$, $\vec{U}$, $\vec{U}_z$, $\vec{U}_r$ $\in$ $\mathbb{R}^{n \times n}$ are weight matrices. m represents the word embedding dimensionality and n represents the number of hidden units, respectively. Logistic sigmoid function is denoted by σ(.).

The backward states $\left(\overleftarrow{h}_1, \dots, \overleftarrow{h}_{T_x}\right)$ are computed likewise where unlike the weight matrices, forward and backward RNNs shares the word embedding matrix $\overline{E}$.

The forward and backward states are concatenated to obtain the annotations $(h_1, \dots, h_{T_x})$ where,

$$h_i = \begin{bmatrix}\vec{h}_i \\ \overleftarrow{h}_i\end{bmatrix} \quad (7)$$

2) Decoder

Given the annotations from the encoder, the hidden state s_i of a decoder is calculates as,

$$s_i = (1 - z_i) \circ s_{i-1} + z_i \circ \tilde{s}_i \quad (8)$$

where,

Proposal hidden state,

$$\tilde{s}_i = tanh(WEy_{i-1} + U[r_i \circ s_{i-1}] + Cc_i) \qquad (9)$$

Update gate,

$$z_i = \sigma(W_z Ey_{i-1} + U_z s_{i-1} + C_z c_i) \qquad (10)$$

Reset gate,

$$r_i = \sigma(W_r Ey_{i-1} + U_r s_{i-1} + C_r c_i) \qquad (11)$$

E symbolizes word embedding matrix for the target language.$W, W_z, W_r \epsilon \mathbb{R}^{n \times m}, U, U_z, U_r\ \mathbb{R}^{n \times n}\ and\ C, C_z, C_r \in \mathbb{R}^{n \times 2n}$ are weights. Again, m and n denotes the word embedding dimensionality and the number of hidden units, respectively. The initial hidden state s_0 is computed as,

$$s_0 = tanh(W_s \overleftarrow{h}_1) \text{ , where } W_s \in \mathbb{R}^{n \times n} \qquad (12)$$

The context vector c_i are recalculated by the alignment model at each step as,

$$c_i = \sum_{j=1}^{T_x} \alpha_{ij} h_j \qquad (13)$$

where,

$$\alpha_{ij} = \frac{exp(e_{ij})}{\sum_{k=1}^{T_x} exp(e_{ik})} \qquad (14)$$

$$e_{ij} = v_a^T\, tanh(W_a s_{i-1} + U_a h_j) \qquad (15)$$

and h_j is the jth annotation in the source sentence. $v_a \in \mathbb{R}^{n'}, W_a \in \mathbb{R}^{n' \times n}\ and\ U_a \in \mathbb{R}^{n' \times 2n}$ weight matrices.

The probability of a target word y_i with the decoder state s_{i-1} , the context vector c_i and the last generated word y_{i-1} is,

$$p(y_i \mid s_i, y_{i-1}, c_i) \propto exp(y_i^T\, W_0 t_i) \qquad (16)$$

where,

$$t_i = [max\, \{\tilde{t}_{i,2j-1}, \tilde{t}_{i,2j}\}]_{j=1,\dots,l}^T \qquad (17)$$

and $\tilde{t}_{i,k}$ is the kth element of a vector $\tilde{t}_i$ which is computed as,

$$\tilde{t}_i = U_0\, s_{i-1} + V_0 Ey_{i-1} + C_0 c_i \qquad (18)$$

$W_0 \in \mathbb{R}^{K_y \times l}, U_0 \in \mathbb{R}^{2l \times n}, V_0 \in \mathbb{R}^{2l \times m}, C_0 \in \mathbb{R}^{2l \times 2m}$ are weight matrices. [1] [2]

B. RNN Encoder – Attention – Decoder Model with Long Short Term Memory

1) Encoder

The bidirectional RNN consists of forward and backward RNN's. The forward RNN $\vec{f}$ intakes the input sequence in order (from x_1 to x_{T_x}) and calculate forward hidden states sequences ($\overrightarrow{h_1}$, …, $\overrightarrow{h_{T_x}}$). The backward RNN $\overleftarrow{f}$ intakes the sequence in the reverse order (from $x_{T_x}\, to\, x_1$) and computes the backward hidden states sequences ($\overleftarrow{h_1}$, …, $\overleftarrow{h_{T_x}}$).

The 1 – of – K coded word vectors from source sentence is taken as input by the model,

$$x = (x_1, \dots, x_{T_x}), x_i \in \mathbb{R}^{K_x} \qquad (19)$$

and outputs as translated sentence of 1 – of – K coded word vectors

$$y = (y_1, \dots, y_{T_y}), y_i \in \mathbb{R}^{K_y}, \qquad (20)$$

where K_x and K_y denotes the source and destination language vocabulary sizes, respectively. T_x and T_y denotes the length of source and destination sentences, respectively.

Firstly, the forward states of the recurrent neural network are calculated as,

$$\overrightarrow{h_\iota} = \begin{cases} o_t\, tanh(c_t) & , if\ i > 0 \\ 0 & , if\ i = 0 \end{cases} \qquad (21)$$

where,

Input gate,

$$\mathrm{I_i} = \sigma(\mathrm{W_i Ex_{i-1}} + \mathrm{U_i h_{i-1}} + \mathrm{Z_i \hat{z}_i} + \mathrm{b_i}) \qquad (22)$$

Output gate,

$$O_i = \sigma(W_0\, Ex_{i-1} + U_0 h_{i-1} + Z_0 \hat{z}_i + b_0) \qquad (23)$$

Memory,

$$c_i = f_i c_{i-1} + I_i\, tanh(W_c\, Ex_{i-1} + U_c h_{i-1} + Z_c \hat{z}_i + b_c)$$
(24)

Forget gate,

$$f_i = \sigma(W_f Ex_{i-1} + U_f h_{i-1} + Z_f \hat{z}_i + b_f) \qquad (25)$$

$E \in \mathbb{R}^{m \times K_x}$ is the word embedding matrix. $U_0, W_0, Z_0, b_0 \in \mathbb{R}^{n \times m}$ are learned weight matrices and biases. m denotes the word embedding dimensionality and n denotes the number of hidden units. σ(.) are logistic sigmoid function.

The backward states $(\overleftarrow{h}_1, \dots, \overleftarrow{h}_{T_x})$ are calculated likewise where unlike the weight matrices, forward and backward RNNs shares the word embedding matrix E.

The forward and backward states are concatenated to obtain the annotations $(h_1, \dots, h_{T_x})$

where,

$$h_i = \begin{bmatrix} \vec{h}_i \\ \overleftarrow{h}_i \end{bmatrix} \qquad (26)$$

2) Decoder

Given the annotations from the encoder the hidden state s_t of a decoder is calculates as,

$$\tilde{s}_t = O_t tanh(\, c_t) \qquad (27)$$

where,

Proposal hidden gate,

$$i_t = \sigma(W_i Ey_{t-1} + U_i s_{t-1} + Z_i \hat{z}_t + b_i) \qquad (28)$$

Output gate,

$$O_t = \sigma(W_0\, Ey_{t-1} + U_0 s_{t-1} + Z_0 \hat{z}_t + b_0) \qquad (29)$$

Memory,

$$c_t = f_t c_{t-1} + i_t\, taanh(W_c\, Ey_{t-1} + U_c s_{t-1} + Z_c \hat{z}_t + b_c) \quad (30)$$

Forget gate,

$$f_t = \sigma(W_f E y_{t-1} + U_f s_{t-1} + Z_f \hat{z}_t + b_f) \quad (31)$$

E $\epsilon\ \mathbb{R}^{m\times k}$ represents the word embedding matrix for the destination language. W_0, U_0, Z_0, b_0 are learned weight matrices and biases. weighted sum is calculated to generate c_t using both precious cell state and current information generated by the cell [3]. m denotes the embedding dimensionality and n denotes the LSTM dimensionality respectively and $\sigma(.)$ be the logistic sigmoid activation. The initial hidden state s_0 is computed as,

$$s_0 = tanh(W_s \overleftarrow{h}_1) \text{ , where } W_s \in \mathbb{R}^{n\times n} \quad (32)$$

Alignment model recomputes the context vector c_i at each step as,

$$c_i = \sum_{j=1}^{T_x} \alpha_{ij} h_j \quad (33)$$

where,

$$\alpha_{ij} = \frac{exp(e_{ij})}{\sum_{k=1}^{T_x} exp(e_{ik})} \quad (34)$$

$$e_{ij} = v_a^T\, tanh(W_a s_{i-1} + U_a h_j) \quad (35)$$

and h_j is the jth annotation in the source sentence. $v_a \in \mathbb{R}^{n'}, W_a \in \mathbb{R}^{n'\times n}$ and $U_a \in \mathbb{R}^{n'\times 2n}$ weight matrices.

The probability of a target word y_i with the decoder state s_{i-1} , the context vector c_i and the last generated word y_{i-1} is ,

$$p(y_i \mid s_i, y_{i-1}, c_i) \propto exp(y_i^T\, W_0 t_i) \quad (36)$$

where,

$$t_i = [max\, \{\tilde{t}_{i,2j-1}, \tilde{t}_{i,2j}\}]_{j=1,\dots,l}^T \quad (37)$$

and $\tilde{t}_{i,k}$ is the kth element of a vector $\tilde{t}_i$, computed as,

$$\tilde{t}_i = U_0\, s_{i-1} + V_0 E y_{i-1} + C_0 c_i \quad (38)$$

$W_0 \in \mathbb{R}^{K_y\times l}, U_0 \in \mathbb{R}^{2l\times n}, V_0 \in \mathbb{R}^{2l\times m}, C_0 \in \mathbb{R}^{2l\times 2m}$ are weight matrices. [4] [1] [2] [5]

VI. Result

In this research work, Attention based Recurrent Neural Network with Long Short Term Memory (LSTM) cells and Gated Recurrent Neural Unit (GRU) cells were implemented for English to Nepali sentence machine translation. The learning rate used for each experiment was 0.001 and 0.2 dropout has been used in this study. The accuracy of the algorithms implemented during this study is determined based on BLEU Score of Test Data.

A. Bilingual Evaluation Understudy (BLEU Score)

It is one of the standard for evaluating the quality of machine-translated text from one natural language to another. The main idea behind the BLEU score is: "the closer a machine translation is to a professional human translation, the better it is" [6]. For each of the translated sentences, scores are determined by comparing them with a collection of high-quality reference translations. The output of BLEU is always a number between 0 and 1. This value illustrates how the candidate text is identical to the reference texts, where values closer to 1 indicate more similar texts. In this study, the Moses muiti-bleu score has be adapted as an evaluation parameter in which the score between 0 to 1 will be converted to the range of 1 to 100. Score 100 indicates a perfect match, whereas 0 indicates a perfect mismatch. The BLEU score is calculated as follows,

$$BP = \begin{cases} 1 & if\ c > r \\ e^{(1-r/c)} & if\ c \le r \end{cases} \quad (39)$$

Then,

$$BLEU = BP.\exp(\textstyle\sum_{n=1}^{N} w_n \log p_n) \quad (40)$$

In the log domain, the ranking behavior is more immediately apparent,

$$\log BLEU = \min\left(1 - \frac{r}{c}, 0\right) + \textstyle\sum_{n=1}^{N} w_n \log p_n \quad (41)$$

Where,

r : the average no. of words in a reference translation, average over all reference translation

c : total length of candidate translation corpus [6]

The BLEU Score for the sample test data set are,

TABLE I. BLEU Score of test data with different layres and hidden units

S.N.	*Number of hidden units*	*Number of layers*	*RNN Unit Type*	*BLEU Score*
1.	128	2	GRU	7.3
			LSTM	6.5
2.	256	2	GRU	12
			LSTM	8.9
3.	512	2	GRU	12.3
			LSTM	11.8
5.	128	4	GRU	1.6
			LSTM	2.6
6.	256	4	GRU	3.1
			LSTM	3.8
7.	512	4	GRU	6.9
			LSTM	3.1

The BLEU score of the test data is calculated using LSTM and GRU cell with attention for different hidden units and number of layers. It is observed that, the best BLEU score obtained is 12.3 with the use of the GRU with attention with 512 hidden layers and 2 layers of neural network.

The translation for a input sentence, "This is merly 6.5 percent of gross domestic product." With best BLUE score using GRU with attention gives output " यो कुल गार्हस्थ्य उत्पादनको ६.५ प्रतिशत मात्र नै मात्र हो". The tag <unk> is used to indicate the unknown words during translation.

VII. Conclusion

English – to – Nepali sentence translation is a task of translating the source English language sentence into the target Nepali language sentence. In this study, recurrent neural network encoder – attention – decoder with GRU cells and with

LSTM cells has been implemented and analysis of the system has been carried out. The English – Nepali parallel corpus has been collected from various sources. The system was trained using eighty percent of data, ten percent of data is used as development set and ten percent of data is used for the testing purpose. The system has been trained with learning rate 0.001 and analyzed with different parameters to test the efficiency of the system. The GRU cells used in encoder and decoder layers with the combination of 512 hidden units and 2 layers of neural network has been able to gain highest BLEU score of 12.3.

References

[1] D. Bahdanau, K. H. Cho, Y. Bengio, "Neural Machine Translation by Jointly Learning to Aligh and Translation," in *ICLR*, 2015.

[2] K. Cho, D. Bahdanau, F. Bougares, H. Schwenk, Y. Bengio, "Learning Phase Representations using RNN Encoder-Decoder for Statistical Machine translation," *arXiv,* vol. 3, no. 1406.1078, 3 9 2014.

[3] H. Hao, B. Li, Z. Qi, W. Shi, J. Tian, B. Xu, P. Zhou, "Attention-Based Bidirectional Long Short Term Memory Networks for Relation Classification," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics* , Berlin, Germany, 2016.

[4] K. Nemkul, S. Shakya, "Low Resource English to Nepali Sentence Translation Using RNN - Long Short Term Memory with attention," in *Proceedings of the international conference on Sustainable Expert Systems(ICSES 2020)*, 2020.

[5] K. Xu, J. L. Ba,R. Kiros, K. Cho, A. Courville, R. Salakhutdinov, R. S. Zemel, Y. Bengio, "Show, Attend and Tell: Neural Image Caption Generation with Visual Attention," in *Proceedings of the 32 nd International Conference on Machine*, Lille, France, 2015.

[6] K. Papineni, S. Roukos, T. Ward, W. Zhu, "BLEU : a Method for Automatic Evaluation of MAchine Translation," in *Proceedings of the 40th Annual Meeting of the Assocaition for Computational Linguistics (ACL)*, Philadelphia, 2002.

[7] T. Young, D. Hazarika, S. Poria, E. Cambria, "Recent Trends in Deep Learning Based Natural Language Processing," *arXiv,* vol. 5, no. 1708.02709, pp. 1-24, 20 02 2018.

[8] Y. Lecun, Y. Bengio, G. Hinton, "Deep Learning," *Nature,* vol. 521, no. 7553, pp. 436-444, 28 5 2015.

[9] E. Greenstein, D. Penner, "Japanese-to-Englisj Machine Translation Using Recurrent Neural Networks," pp. 1-7.

[10] J. Brownlee , "Machine Learning Mastery," 13 10 2017. [Online]. Available: https://machinelearningmastery.com/how-does-attention-work-in-encoder-decoder-recurrent-neural-networks/.

[11] N. J. Khan,W. Anwar, N. Durrani, "Machine Translation Approaches and Survey for Indian Languages".

[12] B. K. Bal, "Structure of Nepali Grammar," *Madan Puraskar Pustakalaya,* pp. 332-396.

[13] R. Agrawal, D. Misra Sharma, "Experiments on different recurrent neural network for English-Hindi machine translation," *CS & IT-CSCP,* pp. 63-74, 2017.

[14] O. Bojar, V. Diatkay, P. Rychlýz, P. Straňák, V. Suchomelz, A. Tamchyna, D. Zeman, "HindEnCorp – Hindi-English and Hindi-only Corpus for Machine Translation," pp. 3550-3555.

[15] B. K. Bal, P. Shrestha, "A Morphological Analyzer and a Stemmer for Nepali," *Madan Puraskar Pustakalaya, ,* pp. 324-331.

[16] B. Keshari, S. K. Bista, "UNL Nepali Deconverter," in *3rd International CALIBER*, Ahmedabad, 2-4 February, 2005,.

[17] "ELRA," ELDA S.A.S, [Online]. Available: http://catalog.elra.info.

[18] [Online]. Available: http://opus.nlpl.eu/.