Algorithms for Intelligent Systems
Series Editors: Jagdish Chand Bansal · Kusum Deep · Atulya K. Nagar

I. Jeena Jacob
Selvanayaki Kolandapalayam Shanmugam
Selwyn Piramuthu
Przemyslaw Falkowski-Gilski *Editors*

Data Intelligence and Cognitive Informatics

Proceedings of ICDICI 2020

Springer

Algorithms for Intelligent Systems

This book series publishes research on the analysis and development of algorithms for intelligent systems with their applications to various real world problems. It covers research related to autonomous agents, multi-agent systems, behavioral modeling, reinforcement learning, game theory, mechanism design, machine learning, meta-heuristic search, optimization, planning and scheduling, artificial neural networks, evolutionary computation, swarm intelligence and other algorithms for intelligent systems.

The book series includes recent advancements, modification and applications of the artificial neural networks, evolutionary computation, swarm intelligence, artificial immune systems, fuzzy system, autonomous and multi agent systems, machine learning and other intelligent systems related areas. The material will be beneficial for the graduate students, post-graduate students as well as the researchers who want a broader view of advances in algorithms for intelligent systems. The contents will also be useful to the researchers from other fields who have no knowledge of the power of intelligent systems, e.g. the researchers in the field of bioinformatics, biochemists, mechanical and chemical engineers, economists, musicians and medical practitioners.

The series publishes monographs, edited volumes, advanced textbooks and selected proceedings.

More information about this series at http://www.springer.com/series/16171

I. Jeena Jacob ·
Selvanayaki Kolandapalayam Shanmugam ·
Selwyn Piramuthu ·
Przemyslaw Falkowski-Gilski
Editors

Data Intelligence and Cognitive Informatics

Proceedings of ICDICI 2020

Springer

Editors
I. Jeena Jacob
GITAM University
Bengaluru, India

Selwyn Piramuthu
University of Florida
Gainesville, FL, USA

Selvanayaki Kolandapalayam Shanmugam
Department of Mathematics and Computer Science
Concordia University Chicago
River Forest, IL, USA

Przemyslaw Falkowski-Gilski
Gdańsk University of Technology
Gdańsk, Poland

ISSN 2524-7565 ISSN 2524-7573 (electronic)
Algorithms for Intelligent Systems
ISBN 978-981-15-8529-6 ISBN 978-981-15-8530-2 (eBook)
https://doi.org/10.1007/978-981-15-8530-2

The ICDICI 2020 conference is solely dedicated to all the editors, reviewers, and authors of the conference event.

Foreword

The International Conference on Data Intelligence and Cognitive Informatics (ICDICI 2020) was held in Tirunelveli, India, during July 8–9, 2020, at SCAD College of Engineering and Technology. The proceedings of ICDICI 2020 conference is presented here with the aim to share and exchange the state-of-the-art research ideas about the different aspects of data and informatics research with a special attention to the encountered practical challenges and the potential solutions adopted to overcome it.

We strongly believe that the research articles of ICIDICI 2020 will give you a technically rewarding experience by providing more research information on the current issues of informatics and general data science interest. IDICI received 247 papers across the country and also from overseas by representing government, industries, and academia out of which 74 papers are duly selected for publication.

ICDICI 2020 promises to be more informative and research simulating with a magnificent array of keynote speakers across the globe. The program consists of invited sessions, presentations, and technical discussion with the most eminent and proficient speaker **Dr. Joy Chen**, Professor, Department of Electrical Engineering, Dayeh University, Taiwan, and session chairs by covering a wide range of topics in data intelligence. Also the conference delegates had a wide range of sessions in different domains of data science, informatics, and cognitive intelligence.

We humbly wish to thank the guest editors **Dr. I. Jeena Jacob, Dr. Selvanayaki Kolandapalayam Shanmugam, Prof. Selwyn Piramuthu, and Dr. Przemyslaw Falkowski-Gilski**, organization staff, technical program committee, and reviewers of the conference for their valuable suggestions and timely response to the authors of

ICDICI 2020. We also extend our gratitude to the authors and conference participants for contributing their novel research results to the conference. Special thanks to Springer publications.

Dr. P. Ebby Darney
Conference Chair
Principal
SCAD College of Engineering
and Technology
Tirunelveli, India

Preface

With a deep satisfaction, I write this Preface to welcome you all to the International Conference on Data Intelligence and Cognitive Informatics (ICDICI 2020) held in Tirunelveli, India, during July 8–9, 2020.

The theme of ICDICI 2020 is *Data Intelligence*, a research topic that is gaining quick research traction in both industries and academia due to its research relevance to the emerging societal and economic issues in the areas like health care, transportation, industries, education, and so on. The well-established research track records on intelligent information systems along with the integration of artificial intelligence techniques and processes make ICDICI an excellent venue for exploring the cognitive foundations in the emerging information systems.

With respect to the potential hardwork of the ICDICI 2020 conference committee, I would like to express my appreciation and gratitude to all the technical program committee members, international and national advisory board members, and review committee members, who have made this conference a successful and possible one.

Finally, I would like to extend my warm thanks to all the keynote speakers, session chairs, and fellow researchers, who have willingly shared their research experience and knowledge to all the readers of this extended conference proceedings.

I hope that this proceedings of ICDICI 2020 will further stimulate research in data mining and intelligent systems and provide practitioners with advanced algorithms, techniques, and tools for deployment. I feel honored and privileged to serve the significant recent developments in the field of intelligent systems and data intelligence to you through this exciting program.

Bengaluru, India

Dr. I. Jeena Jacob
ICDICI-2020

Acknowledgements

We are deeply obliged to all the contributors of this ICDICI 2020 conference and all the reviewers for their informative, cogent, and timely reviews of papers submitted to the conference, and also the SCAD College of Engineering and Technology staffs and international technical reviewer's community with their professional and thoughtful expertise to shape this conference event.

Overall thanks go to all the conference programme and local organizing committee members, who have gone out in their way to make this as a successful event.

Finally, we offer our sincere thanks to conference chair, co-conference chair, and organizing secretary for their continuous effort in organization, preparation, and handling of ICDICI conference administration. Further appreciation is also due to the editors of Springer publications, high standards of editorial productions of conference proceedings.

Contents

1 **Text Generation and Enhanced Evaluation of Metric for Machine Translation** 1
Sujit S. Amin and Lata Ragha

2 **Destress It—Detection and Analysis of Stress Levels** 19
Neha Udeshi, Nemil Shah, Urvi Shah, and Stevina Correia

3 **On the Design and Applications of an Integrated Smart Home Automation** 35
Thanaz Shafeer, M. Senthil Arumugam, and A. Sasikala

4 **Voice and Gesture Based App for Blind People** 53
Tilak Satra, Manali Shah, Ajay Lad, and Stevina Correia

5 **Digital Image Forgery Detection Using Ternary Pattern and ELM** 77
D. Vaishnavi, D. Mahalakshmi, and M. S. Shawon Rahman

6 **A Study on Identification and Cleaning of Struck-Out Words in Handwritten Documents** 87
R. Dheemanth Urs and H. K. Chethan

7 **RFID-Based Railway Crowd Prediction and Revenue Analysis** . . . 97
Shraddha S. More, Gulshan Pathak, Simran Panchal, and Monika Patil

8 **A Review on Efficient Scheduling Techniques for Cloud Computing** 111
Venkateswara Rao Kumpati and Manish Pandey

9 **A Synonym-Based Bi-LSTM Model for Machine Reading Comprehension** 121
S. Anbukkarasi, N. Abinaya, and S. Varadhaganapathy

10 Designing of Arbiter PUF for Securing IP and IoT Devices 131
Swati Kulkarni, R. M. Vani, and P. V. Hunagund

11 Enhancements on the Efficacy of Firefighting Robots Through Team Allocation and Path-Planning 139
Sreesruthi Ramasubramanian, Senthil Arumugam Muthukumaraswamy, and A. Sasikala

12 Smart Meter Using Raspberry Pi for Efficient Energy Utilization 153
Chidrupa Manogna Mamunooru, Suseendra Attada, Sai Vipanchi Kodali, Mohan Reddy Tumma, Santanu Kumar Dash, and Y. Padma Sai

13 An Unsupervised Content-Based Article Recommendation System Using Natural Language Processing 165
S. Renuka, G. S. S. Raj Kiran, and Palakodeti Rohit

14 A Survey of the Exemplary Practices in Network Operations and Management 181
K. M. Majidha Fathima

15 Utilization of Fish Excrete for Plant Growth Using SVM 195
C. Gnana Kousalya, T. U. Gowdhami, and R. Mayavinothini

16 A Novel Approach of Resource Scheduling Algorithm to Improve QoS in Green Cloud Computing 207
P. Geetha and Dr C. R. Rene Robin

17 Machine Learning Approach for Analysis of the Company Performance Based on Fundamental Data 223
Pratik Rathi, Madhur Kabra, Rahil Sheth, and Jignesh Sisodia

18 Evaluation of Deep Learning Models in the Prediction of Lung Disease(Pneumonia) 233
Adusumilli Rohit, B. Padmaja, K. Vinay Kumar, T. Chandana, and M. Madhu Bala

19 An Automated Invoice Handling Method Using OCR 243
Pranay Kumar and Dr. S. Revathy

20 Performance Evaluation of Clustering-Based Classification Algorithms for Detection of Online Spam Reviews 255
N. Krishnaveni and V. Radha

21 Product Recommendation Systems Based on Customer Reviews Using Machine Learning Techniques 267
J. S. Shyam Mohan, Hanumath Sreeman Vedantham, Venkata Chakradhar Vanam, and Nagendra Panini Challa

22 Mobile Application for Classification of Plant Leaf Diseases Using Image Processing and Neural Networks 287
K. S. Chethan, Sumanth Donepudi, H. V. Supreeth, and Vachan D. Maani

23 Energy-Efficient System-Based Algorithm for Maximal Resource Utilization in Cloud Computing 307
Arvind Dagur, Mishra Adityakumar Virendra, Mayank Pratap Singh, Deepak Gupta, Chetanya Sharma, and Rahul Chaturvedi

24 Energy Enhancement of WSN Using Fuzzy C-Means Clustering Algorithm 315
Arvind Dagur, Nidhi Malik, Priyanshi Tyagi, Rishu Verma, Riya Sharma, and Rahul Chaturvedi

25 Optimization of Queries in Database of Cloud Computing 325
Arvind Dagur, Ankit Kaushik, Adamya Rastogi, Amritpal Singh, Ashish Kumar, and Rahul Chaturvedi

26 Low-Cost IoT-Enabled Smart Parking System in Crowded Cities 333
Rahul Chaturvedi, Shubham Kumar, Utkarsh Kumar, Tanisha Sharma, Zeba Chaudhary, and Arvind Dagur

27 A Critical Review on Educational Data Mining Segment: A New Perspective 341
Randhir Singh and Saurabh Pal

28 Exploration and Visualization of the Hidden Information from the Congestive Heart Failure Patients Data in MIMIC-III Database 349
S. Gayathri, M. Anitha, S. Nickolas, and S. Mary Saira Bhanu

29 Neuromemorize—Mobile Application for Alzheimer's Patients Using AI-Based Face Recognition 363
Aziz Presswala, Srushti Pathak, Gunjan Jhanwar, and Neha Katre

30 Brain Tumor Classification Using PCA and PNN of T1 and T2 Weighted MRI Images 379
Bhakti Gangan, Manjusha Deshmukh, and Dhananjay Borse

31 Workload-Driven Transactional Partitioning for Distributed Databases 389
R. D. Bharati and V. Z. Attar

32 Start-Up Leagility Assessment Using Multi-grade Fuzzy and Importance Performance Analysis 397
Edrion Chacko, M. Suresh, and S. Lakshmi Priyadarsini

33 Assessment of Leagility in Healthcare Organization Using Multi-grade Fuzzy Approach 409
V. Vaishnavi and M. Suresh

34 A Unified Model for Face Detection Using Multiple Task Cascaded Neural Network Prepended with Non-local Mean Denoising Algorithm 423
Sapna Rathore and Luv Sharma

35 Workplace Stress Assessment of Software Employees Using Multi-grade Fuzzy and Importance Performance Analysis 433
S. Sreedharshini, M. Suresh, and S. Lakshmi Priyadarsini

36 Early Detection of Mild Cognitive Impairment Using 3D Wavelet Transform 445
B. A. Sujatha Kumari, A. G. Varun Yadiyala, B. J. Aruna, C. Radha, and B. Shwetha

37 Spatial Data Mining of Agricultural Land Area Using Multi-spectral Remote-Sensed Images 457
Parminder Kaur Birdi, Karbhari Kale, and Varsha Ajith

38 Person Identification Using Histogram of Gradient and Support Vector Machine on GEI 471
P. Nithyakani and M. Ferni Ukrit

39 Inventory Optimization for Cognitive Demand Scheduler Using Data Analytics 479
Priyanka Singh, Soma Ghosh, Manish Saraf, and Rahul Nayak

40 DRIVE SAFE: Lane Deviation Detection and Alert System Using Image Processing Techniques 495
H. C. Arun, Jisha John, and Aswathy Ravikumar

41 Foodborne Illness Investigation Technique Using Machine Learning 507
Ahmad Alkinj, Shabina Ghafir, M. Afshar Alam, and Md Tabrez Nafis

42 A Primer on Word Embedding 525
Satvika, Vikas Thada, and Jaswinder Singh

43 Automatic Cataract Detection Using Haar Cascade Classifier 543
Jaspreet Kaur, Prerit Sinha, Rahul Shukla, and Vikas Tiwari

44 VG2—DNA-Based One-Time Pad Image Cipher 557
Akhil Kaushik, Vikas Thada, and Jaswinder Singh

45 Hand Gesture Recognition Control for Computers Using Arduino 569
J. S. Vimali, Senduru Srinivasulu, J. Jabez, and S. Gowri

46 Infrared Small Target Detection Based on Fractional Directional Derivative and Phase Fourier Spectrum Transform 579
Sur Singh Rawat, Sashi Kant Verma, and Yatindra Kumar

47 Detection of Pre-term Delivery by the Analysis of Fetal ECG Signals 593
P. Sridharan and V. Dhileep

48 Sentiment Analysis of Twitter Data Using Techniques in Deep Learning 613
S. Gowri, J. Jabez, J. S. Vimali, A. Sivasangari, and Senduru Srinivasulu

49 Part-of-Speech Tagging in Mizo Language: A Preliminary Study 625
Morrel V. L. Nunsanga, Partha Pakray, Mika Lalngaihtuaha, and L. Lolit Kumar Singh

50 Phishing Website Prediction: A Comparison of Machine Learning Techniques 637
Anjaneya Awasthi and Noopur Goel

51 Comparison of Computer Vision Techniques for Drowsiness Detection While Driving 651
Deepanshu Yadav, Divya Mohan, and Amrita Jyoti

52 Machine-Learning Based Fault Diagnosis of Electrical Motors Using Acoustic Signals 663
Ravi Teja Grandhi and N. Krishna Prakash

53 Study of Users Attitude and Classification of Comments and Likes from Facebook Using RapidMiner 673
Mohammad Kashif Khan and Nafisur Rahman

54 Novel State Disturbance Based Multi-level Inverter with Sliding Mode Control 689
Telma Johnson and V. Shijoh

55 Survey on Graphical Password Authentication System 699
Shikhar Singh Patel, Akarsh Jaiswal, Yash Arora, and Bharti Sharma

56 Predictive Analysis for Early Detection of Alzheimer's Disease . . . 709
B. A. Sujathakumari, M. Charitha Shetty, H. M. Lakshitha, P. Jain Mehulkumar, and S. Suma

57 Feature Extraction from Ensemble of Deep CNN Model for Image Retrieval Application 725
Vijayakumar Bhandi and K. A. Sumithra Devi

58 Performance Analysis on Machine Learning Algorithms with Deep Learning Model for Crop Yield Prediction 739
Supreetha A. Shetty, T. Padmashree, B. M. Sagar, and N. K. Cauvery

59 A Novel Hybrid Cipher Using Cipher Squares for Better Security 751
Adilakshmi Visali Jayanthi

60 A Review on Dynamic Virtual Machine Consolidation Approaches for Energy-Efficient Cloud Data Centers 761
B. Prabha, K. Ramesh, and P. N. Renjith

61 A Secure Video Steganography Using Framelet Transform and Singular Value Decomposition 781
Meenu Suresh and I. Shatheesh Sam

62 Collective State Implementation on Particle Swarm Opimization for Feature Selection 791
Jaswinder Singh, Soham Pathak, and Ritwik Bandyopadhyay

63 Enhanced Dragonfly Algorithm Adapted for Wireless Sensor Network Lifetime Optimization 803
Miodrag Zivkovic, Tamara Zivkovic, K Venkatachalam, and Nebojsa Bacanin

64 Detection of Pneumonia Clouds From Chest X-ray Images 819
L. Suganthi, K. Nirmala, S. Deepa, K. Nagalakshmi, and M. Santhya

65 Effectiveness of Blockchain Advancement in Patient Statistical Monitoring Network 831
M. Malathi, R. S. Krupasree, T. Lalitha Bhuvaneshwari, and S. Gokulraj

66 Generation of Alternative Clusterings Using Multi-objective Particle Swarm Optimization 841
Avinash Navlani and V. B. Gupta

67 Region of Interest-Based Encryption of Biomedical Image 851
Srinivas Vodnala, Shubhankar Majumdar, and Pallab Kumar Nath

68 Implementation of Production Planning Using Overall Equipment Efficiency (OEE) for IIoT 863
K. Amar Prem

69 Counting and Tracking of Vehicles and Pedestrians in Real Time Using You Only Look Once V3 873
O. Swetha and C. Ramachandran

70 Comprehensive Analysis of Multimodal Recommender Systems . . . 887
Viomesh Kumar Singh, Sangeeta Sabharwal, and Goldie Gabrani

71 All-Zero Pre-processing Over ISI Channels: A Methodology for TCM Schemes 903
Vanaja Shivakumar

72 New Technical Using Nano in Medical Field to Determine Medications that are Suitable Activities for COVID-19 917
Shuker Mahmood Khalil and Nadia M. Ali Abbas

73 Usability Evaluation of Virtual Reality-Based Fire Training Simulator Using a Combined AHP and Fuzzy Comprehensive Evaluation Approach 923
El Mostafa Bourhim and Abdelghani Cherkaoui

74 Study the Basic Survey of Food Quality Parameter Analysis Technique Using Image Development Methodology 933
Sanket Mungale and Deepak S. Khot

Author Index 939

About the Editors

I. Jeena Jacob is working as a Professor in Computer Science and Engineering department at GITAM University, Bangalore, India. She actively participates on the development of the research field by conducting international conferences, workshops and seminars. She has published many articles in refereed journals. She has guest edited an issue for the International Journal of Mobile Learning and Organisation. Her research interests include mobile learning and computing.

Selvanayaki Kolandapalayam Shanmugam is currently working as an Associate Professor in Computer Science, Concordia University, Chicago, USA. She had over all 15+ years of lecturing sessions for theoretical subjects, experimental and instructional procedure for laboratory subjects. She presented more research articles in the national & international conferences and journals. Her research interest includes image processing, video processing, soft computing techniques, intelligent computing, web application development, object-oriented programming like C++, and Java, scripting languages like VBscript and Javascript, data science, algorithms, data warehousing and data mining, neural networks, genetic algorithms, software engineering, software project management software quality assurance, enterprise resource planning, information systems and database management systems.

Selwyn Piramuthu is Professor of Information Systems at the University of Florida. He received his B.Tech., M.S. and Ph.D., respectively, from IIT Madras, University of Arizona and University of Illinois at Urbana–Champaign. His research interests include machine learning and cryptography with applications in medical informatics, supply chain management, financial credit risk scoring, IoT, among others. His book, co-authored with Wei Zhou titled, "RFID and Sensor Network Automation in the Food Industry," was published by Wiley in 2016.

Przemyslaw Falkowski-Gilski is a graduate of the Faculty of ETI, Gdansk University of Technology. He graduated 1st degree B.Sc. studies (in Polish) and 2nd degree M.Sc. studies (in English) in 2012 and 2013, respectively. He pursued his Ph.D. studies in the field of electronic media, particularly digital broadcasting

systems and quality of networks and services. In 2018, he received the title of Doctor of Technical Sciences with distinction, discipline Telecommunications, specialty in Radio communication. Currently, he works as an academic.

Chapter 1
Text Generation and Enhanced Evaluation of Metric for Machine Translation

Sujit S. Amin and Lata Ragha

1 Introduction

Recurrent neural networks (RNNs) have the capability to learn from the given text and generate new text or translate the given text. They are powerful since they have a lot of hidden states with nonlinear dynamics that permit them to process them previous information. Further, gradients of RNN are cheap to compute. Even with all these features, RNNs are difficult to train properly. The cause is the "vanishing gradient problem" [1]. Vanishing gradients problem occurs when neural networks have many layers in neural network gradients of loss functions approach to zero, making it hard to train. Because of this, there have been limited successful applications using RNNs [2].

Gated recurrent units help to fine-tune neural networks input weights to resolve vanishing gradient problem. Gated recurrent unit (GRU) is a variant of long short-term memory (LSTM) because both have a similar design; GRUs are an improved version of the recurrent neural network [3]. Gated recurrent unit is known to have good text generation in many languages [4].

Our model uses GRU architecture. Our model uses a sequence memorizer [5], which is a hierarchical nonparametric Bayesian method. The memorizer brings dependencies between its predictions by making similar predictions at related contexts.

S. S. Amin (✉) · L. Ragha
Department of Computer Engineering, Fr. C. Rodrigues Institute of Technology, Vashi, Navi Mumbai, India
e-mail: mastersujitamin@gmail.com

L. Ragha
e-mail: lata.ragha@gmail.com

I. Jeena Jacob et al. (eds.), *Data Intelligence and Cognitive Informatics*,
Algorithms for Intelligent Systems,
https://doi.org/10.1007/978-981-15-8530-2_1

The problem is identified that the text generated by RNN might not be grammatically correct, GRU, although learns patterns after training for a large number of times. At times, it does make grammatical mistakes. To combat that problem, the grammatical correction model is used using natural language processing (NLP) at the end of text generation. Grammar model uses language modelling, word tagging to check grammar for generated text sentence by sentence. To achieve this, natural language toolkit (NLTK) is used. From framework, it uses part-of-speech tagging, noun phrase extraction, spelling corrections, tokenization and sentiment analysis to achieve grammar auto-corrections.

Text generated after grammar corrections was a significantly high level of linguistic structure and has a considerable amount of grammatical structure in it.

Next component is machine translation (MT), and MT is a field that uses machines to translate text from language to language. During translation, the meaning of the text must be similar in the goal language. The translation is not a replacement of word for the word problem.

In [6], the authors talk about the neural machine translation (NMT) system, which is applied in English to German. Authors have achieved a state-of-the-art model for NMT. In [7], the authors achieved multiple translations of the same paragraphs. Authors have achieved high-quality translations. Translation system uses LSTM networks.

In MT, the sentence-level score is calculated by the BLEU metric. The bilingual evaluation understudy (BLEU) metric is a metric developed by IBM, and it is a way to measure translation quality [8]. Closer the translation to an expert human translation, the superior it is [8]. But BLEU metrics have drawbacks when evaluating Indian languages. So, this drawback aimed to remove using modified BLEU metric for translation. Modified BLEU uses synonym replacing algorithm and after that shallow parsing algorithm was used to get better results (Fig. 1).

This way, two goals are accomplished: one is grammatically correct generated English text from text generator and improved BLEU score for translation.

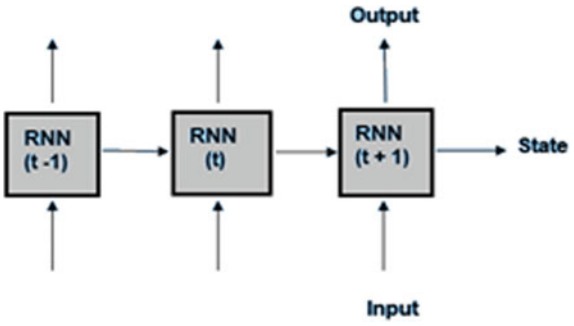

Fig. 1 RNN is a very deep feed-forward network which uses nonlinear activation function

2 System Overview

The system's goal is to generate English text from a given input and then translate text to Hindi with high translation score (BLEU score). As shown in Fig. 2, the system first takes the input and then generates text in utf-8 format. UTF-8 format is chosen because it is difficult to convert text into Hindi with another encoding format. Input can be a word, sentence or paragraph. Automatic text generation module contains a recurrent neural network (RNN) for creating text from the input text. Since automatic text generation module output can be grammatically incorrect. So, the grammar correction module contains NLTK is used for correcting the grammar text sentence by sentence. Now, got our first output English text. This output is given to the English to Hindi translation module. English to Hindi translation module contains RNN for translation. Now a BLEU score is got which measures the translation accuracy. The BLEU score has known to be less for English to Hindi translation since Hindi is a morphologically rich language [9]. So, the modified BLEU score module is used, which uses synonym replacement and shallow parsing to overcome this issue.

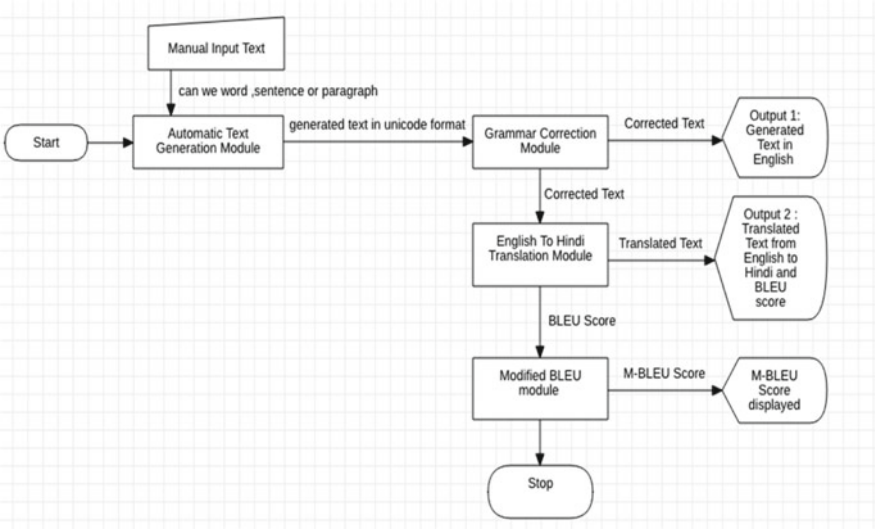

Fig. 2 Flow chart for the overall working of text generation and translation system

3 Neural Networks

3.1 Recurrent Neural Networks

A recurrent neural network is a feed-forward neural network that allows to model sequential data [4]. At each time step, RNN loads an input, revises its hidden state and makes a prediction (Fig. 1). The RNN has a high-dimensional network and has nonlinear activation function over many time steps that enable them to have accurate predictions.

Activation functions are computational equations, which dictate a neural network's output. The function is linked to each neuron in the network and decides if it should be triggered ("fired") or not, based on whether the input of each neuron is essential to the evaluation of the model. Activation functions also help to normalize each neuron's output to a range of around 1 and 0 or −1 to 1.

The standard RNN is written as follows: given a series of input vectors $(x_1, \ldots, x_T)$, the RNN calculates the hidden state sequence $(h_1, \ldots, h_T)$ and the output sequence $(o_1, \ldots, o_T)$ by iterating the following equations for $t = 1$ to T [4].

$$h_t = \tanh(W_{hx}x_t + W_{hh}h_{t-1} + \mathrm{b_h}) \tag{1}$$

$$o_t = W_{oh}h_t + b_o \tag{2}$$

In the above equations, W_{hx} is input, hidden node to hidden node weight matrix is W_{hh}, hidden node to the output node weight matrix is W_{oh} and the vectors b_o and b_h are bias [4]. The undefined expression $W_{hh}h_{t-1}$ at time $t = 1$ is changed with a bias vector $hinit$ and $tanh$ is applied coordinate-wise.

The gradients of RNN are computationally easy to calculate, so maybe concluded that RNNs are simple to train with gradient descent [10]. In reality, relationship and dynamics are unstable. Moreover, gradient decays exponentially through time. Moreover, the backpropagated gradient's occasional tendency improves the gradient descent and makes it very unstable. To deal with this problem is to change the standard RNN structure so that it can store information over a long time period. This is known as "long short-term memory" (LSTM) [11]. It outperformed the standard RNN. But LSTM is computational and memory intensive. LSTM is also less efficient. A better approach is gated recurrent unit (GRU) [3]. GRU is a simpler version that uses two gates, easier to modify and does not need memory units. Moreover, GRU is quicker to train than LSTM. LSTM is good for a machine translation system.

3.2 Gated Recurrent Unit

GRU is a recently proposed method in [3] which allows acquiring dependencies of various timescales. LSTM is similar to GRU. GRU has several gating units that allow the flow between data. GRU is good as a generative model. GRU performs better than LSTM [3].

4 Various Models of RNN

4.1 RNN as Generative Model

The goal of text generative model is predicting the upcoming word in the sequence. GRU is used for the generative model since it is computational efficient and uses less memory. GRU has two gates. GRU does not have internal memory. Also, the nonlinearity model did not apply and clearly, much better than LSTM. Standard RNN uses replace activation with a new value calculated from current input and hidden state [3]. GRU keeps the available content and adds unique content on top of it. This enables conditional distribution in a produced string to get the next character and provide the next input to RNN. Thus, the model is directed non-Markov model; it resembles the sequence memorizers [4].

4.2 Natural Language Processing for Grammar Correction

The goal of grammar correction is to understand if the grammatical error is present and auto-corrects the sentence [12]. This uses NLP to accomplish the objective. This is used since generated text may not be grammatically correct. NLP tool uses tokenization to convert a sentence into tokens. Then language is detected. Once tokenization and language check are done, spelling check is done. This is done to avoid spelling errors. Finally, the model checks for any semantics error in every sentence and checks for what sentence contains (numbers, alphabet and punctuation). After checking if incorrect text (numbers, alphabet, punctuation) is present, then it is replaced with the correct word or remove text.

4.3 RNN as Translation Model

Translation can be done using neural machine translation (NMT) [7]. NMT has produced a state-of-the-art outcome for English to French, English to German and English to Czech [7]. The goal of NMT is to model directly source sentence (English),

to a target sentence (Hindi). It accomplishes by encoder–decoder framework [7]. For each source sentence, the encoder calculates a depiction [13]. The decoder produces an output, one word at a time, based on that depiction of the source (see Fig. 3). By taking a weighted mean of hidden states, the context vector is determined. Figure 4 describes the attention mechanism used in our translation system. Translation model uses LSTM networks.

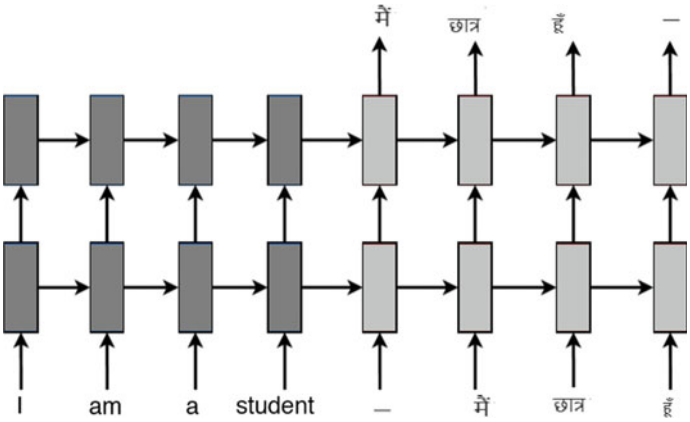

Fig. 3 Neural machine translation—here is an example of RNN for translating English ("I am a student") to ("मैं छात्र हूँ") Hindi [7]

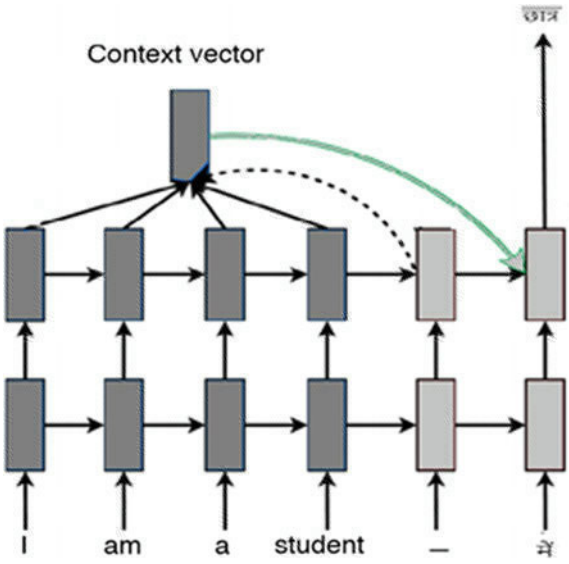

Fig. 4 Attention mechanism—mechanism has two steps: firstly, context vector is calculated by using hidden state; secondly, context vector is used to calculate next hidden state [2]

5 Translation Assessment

Translation assessment answers the accuracy of the translation. Initially, MT assessment was done by humans. It used to be subject from person to person and time-consuming. Assessment varies from one evaluator to another. Hence, the need for automatic translation system is developed. BLEU is the most popular technique developed for the automatic translation system. Human assessment cannot be used every time due to several reasons. On a broader perspective, it has a human assessment and automatic assessment.

5.1 Human Assessment

For human assessment, a speaker of the translated language (e.g. Hindi) and source language (e.g. English) [14] is needed. Human evaluator's translation quality is influenced by various factors such as fluency and adequacy while assessing translation. Moreover, the process can take weeks to months to complete. Monitoring cost also adds up to the total cost. Due to the above variables, there is a need for an automatic assessment scheme

5.2 Automatic Assessment

If the assessment of text is done by machines, then it is automatic machine translation assessment. There are two levels of evaluation. First is sentenced level, and an algorithm is run here on a bunch of sentences then compared against human judgement. Secondly, having a corpus level, the score is calculated by assembling the score of both human and metric judgements. Finally, the aggregate score is calculated. The benefit of automatic machine assessment is that it is much faster.

6 Measuring of Automatic Assessment

The metrics for the MT system are the accuracy of the translated output. However, the quality is subjective. So, metric responsibility is to get the score for translated output text. The score should match with human judgement. Therefore, the metric should give a high score for translations where human will give a high score and low score to translations where humans will give low scores. The BLEU metric has been used for our system.

6.1 BLEU Metric

The bilingual evaluation understudy (BLEU) metric is a metric developed by IBM [8]. The concept of BLEU is nearer a translation output to a translation done by a human, superior the output. Scores are computed for unique segments that are translated. This is usually done by phrasing and comparing translated segments with a set of reference translations of good quality. Those scores are then weighted around the entire corpus to get an estimation of the perceived quality of the translation. No account is taken of the fluency or grammatical correctness. The output of BLEU is always a number between 0 and 1. *N*-gram comparison is used for checking the translation accuracy.

N-gram concept is used in the BLEU metric [8]. The difference is a check between both translations. The score is calculated by the formula:

$$\text{BLEU} = \text{BP} * \exp\left[\sum_{n=1}^{N}\left\{\left(\frac{1}{N}\right) * \log(P_n)\right\}\right] \tag{3}$$

Here N is max n-gram size. n varies from 1 to N [8].

7 Issues with BLEU

BLEU is the most popular method for evaluation of the translated text. But while evaluating English to Indian dialects, it has many disadvantages. [9]. The significant BLEU disadvantages in automatic English-Indian language evaluation are only discussed.

- Poor correlation with judgements of humans.
- The same term in two different forms is considered to be distinct.
- BLEU considers similar meaning words or about the same meaning words as different words.
- The better score does not indicate better translation.
- Many references are required for evaluation purpose.

8 M-BLEU Module

As shown in Fig. 5, the grammatically correct generated text is fed to the machine translation engine. MT engine is two layers deep RNN network for translation. From that, English to Hindi is translated. In the root word extraction part, morphological analysis is done, i.e. is to split the sentence in morphemes. The lexicon contains the stock of words. Lexicon is like a dictionary. Synonyms are grouped. To make correct sentences after synonym replacement process, shallow parsing is used. Then

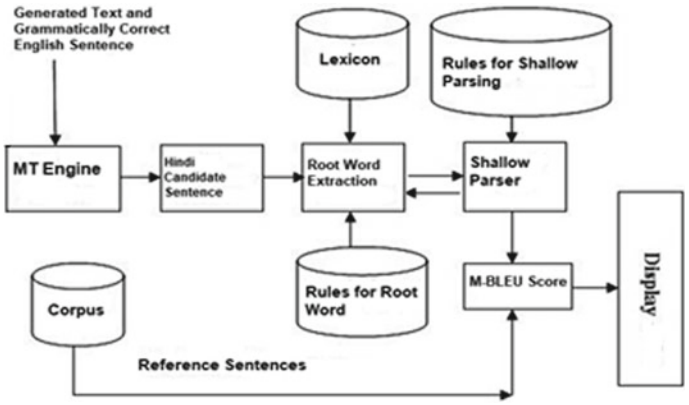

Fig. 5 Modified BLEU module overall working

reference sentences are checked and generated sentences for BLEU score. Shallow parsing and synonym replacement are done several times for every sentence. Until the best BLEU score is reached. Finally, M-BLEU score is calculated. For the entire text, M-BLEU score is calculated by taking the mean of M-BLEU score.

9 Experiments

The goal of our experiments is to demonstrate first that grammatically correct text can be generated from RNN when the gated recurrent unit is used. Second, generated text can be converted to Hindi text with better BLEU score by removing drawbacks.

9.1 Text Generation and Grammar Correction of Generated Text

Aim of this part is to generate text using RNN using the gated recurrent unit. This is demonstrated by comparing LSTM and GRU on English language data sets. Owing to nonparametric nature, LSTM used a large amount of memory while training. GRU, on the other hand, used considerably less memory. The GRU and LSTM are trained on the same size of the data set. Finally, the grammar correction part is applied which gave good correction output text on both LSTM and GRU.

Data sets
The data set is now described. The data set contains text with 80 different types of utf-8 characters which include punctuations, digits and special characters. English text here used is American English. The books have been excluded from other English,

such as Indian English and British English. Data set is made from a collection of books. Ten books are used to make data set. Books were of pdf format. It was converted to plain text. Every book was cleaned by removing any page number, unnecessary line breaks and headers. Any XML and markup were removed to clean the data set. Greek letters, Latin letters and formula were removed since its output is the English text. Punctuations were retained without punctuations output looks obscure. The book's text is randomly permuted. Data set was about 140 MB.

Details

To process the training set (Eq. 2), the entire data set is needed to process, which is infeasible because of the size of the data set system ran out of memory while training. Training the LSTM or GRU can also be done on shorter sequences of character which is just as effective provided they are hundred characters or longer. Advantage of using relatively short sequence is that it is easy to parallelize. Three parallel high-end GPUs and 6 GB of RAM each are used.

Firstly, our model reads the text file and then splits the content character by character. Then characters are sorted. Count of unique characters is also taken. Mapping is done from a unique character to indices. Another index to character mapping is done. Both are used while training. Sequence length is kept to optimal value so that text generated is effective. Data set is taken in slices and run parallel on different GPUs. By looking at the present character, the model predicts the next character. The batch size was 64. The model has 1200 hidden nodes. Small buffer is maintained so that shuffling can be done. It does not shuffle the entire sequence in memory. Model is sequential.

The model has been implemented under the Python programming language. Keras and TensorFlow Library have been used. Keras is the API for neural networks. TensorFlow is a digital computing library. It is devoted to the study of machine learning.

Grammar correction uses natural language toolkit (NLTK). Grammar correction is used since the text generated from the GRU and LSTM recurrent model might not be grammatically correct to sound. Grammar correction uses NLP tools. First, tokens are generated from generated text. Tokens are generated from word to word format. A character-level token does not fit for grammar correction. Then language is detected using spaCy. spaCy is a framework used for NLP. spaCy makes it simple to detect languages. Next, NLTK is used for spelling checks. Spelling checks remove any spelling errors in the text. Next part, check for semantic error; this is done sentence by sentence. Semantic is checked sentence by sentence. Semantic information of text is checked for, i.e. what the text contains (numbers, alphabet, punctuation). After understanding the semantics, if the incorrect word (numbers, alphabet, punctuation) is present, then replaced with the correct word or remove a word.

To validate our approach of generated text, two cases are there by our proposed method.

- LSTM applied to the text: standard LSTM architecture is used on the text.
- GRU applied to the text: as originality of the paper, GRU is used with grammar correction.

Experiment condition is same three parallel high-end GPU, 6 GB of RAM each. A utf-8 encoding has been used. The model took 2 weeks to train properly. The multinomial distribution is used to predict word returned by model. As training progresses, training loss was reduced. Every 100 epochs are gathered in the checkpoint file. Checkpoint file is maintained so that it does not lose model if, for some reason, the machine stops processing. The best set weights are used (least loss) to our generative model.

9.2 Text Translation and Modified BLEU Score for Translation Accuracy

Aim of this module is to translate text from the output of text generation module to Hindi language and generate better BLEU score. The NMT has been used for machine translation. Basically, RNN has been used for the translation since RNN has shown to be very useful in text translation [6].

9.3 Training Details

The data set is now described. Data set is a collection of sentence pairs in English and Hindi which is also made by a collection of books contained both Hindi and English versions. Then the text was carefully placed in pairs. UTF-8 encoding was used. The data set contains about 3 million sentence pairs. Vocabulary is made limited to 50 thousand frequent words. Other words are represented as <unk>. 20% of the data was for validation and test set.

Two layers of LSTM are used with 1000-dimensional embedding. Other methods were applied like attention mechanism and source reversing [15]. The model was trained on the GPU. BLEU score is generated for checking accuracy for translation.

The original BLEU scores were low. As Hindi is a morphologically rich language, so, the modified BLEU (M-BLEU) score is created. M-BLEU module contains shallow parsing and synonym replacement on a different set of sentences. This resulted in better BLEU score as compared to old BLEU score. Synonym data is created from available Hindi data set of synonym. Synonym replacement contains synonyms for every Hindi word, and synonyms of every word are replaced.

At the same time, wrong semantics might be encountered for every sentence where synonym is replaced, and shallow parsing is used. Shallow parsing is basically checked for semantics after replacement if it is correct or not. If incorrect, the sentence is reframed. Once reframing is done, M-BLEU score is checked for.

This works because synonym is considered a different word while calculating BLEU; whereas, while calculating modified BLEU, synonyms are replaced by several