Cloud-based Sanskrit to Hindi Machine Translation System

A

Thesis

submitted for the award of degree of

DOCTOR OF PHILOSOPHY

by

Muskaan Singh
(951503012)

Under the guidance of

Dr. Ravinder Kumar
Associate Professor

Dr. Inderveer Chana
Professor and Dean, Student Affairs

Computer Science and Engineering Department
Thapar Institute of Engineering & Technology(TIET),
Patiala -147004, India

May 2021

Contents

List of Figures vii
List of Tables ix
Certificate xiii
Acknowledgments xv
Abstract xvii

1 Introduction **1**
1.1 Machine Translation:An Overview 2
1.1.1 Machine Translation Approaches 3
1.1.2 Properties of Machine Translation System 6
1.1.3 Applications of Machine Translation 6
1.1.4 Need for Machine Translation Systems 7
1.1.5 Characteristics for Sanskrit Language Processing 8
1.2 Cloud Computing: An Overview 9
1.2.1 Cloud Service Models 11
1.2.2 Cloud Deployment Models 11
1.2.3 Cloud Computing Components 12
1.2.4 Need of Machine Translation Deployment on Cloud 13
1.3 Problem Statement 14
1.4 Research Motivation 16
1.5 Objectives of the Proposed Work 16
1.6 Thesis Contribution 17
1.7 Thesis Organization 19

2 Literature Survey **23**
2.1 Machine Translation Modelling Techniques 24

2.1.1 Human-engineered Translation 24
2.1.2 Machine-engineered Translation 27
2.1.3 Hybrid Machine Translation(HBMT) Modelling 31
2.2 Challenges of MTS for Processing Sanskrit Language 37
2.2.1 Linguistic Challenges 37
2.2.2 Technical Challenges 38
2.3 Comparison of Various Modelling Techniques 41
2.4 Comparison of Machine Translation Systems 43
2.5 Machine Translation System for Processing Sanskrit and Hindi Languages 46
2.6 Existing Machine Translation System on Cloud 48
2.7 Conclusion 49

3 Proposed Sanskrit-Hindi Hybrid Machine Translation System 55
3.1 Proposed MTS Description 56
3.1.1 Data Pre-processing of Corpus 56
3.1.2 Rule-based Machine Translation System for Extracting Linguistic Features 56
3.1.3 SHH-MTS: Neural Network-based RNN Approach 58
3.1.4 SHH-MTS: Hybrid Approach 65
3.2 Experimental Design 65
3.2.1 Corpora 65
3.2.2 Model Size 72
3.2.3 Parameter Initialization 72
3.2.4 Training 72
3.3 Performance Evaluation 73
3.3.1 Automatic Error Analysis 75
3.3.2 Human Error Analysis 79
3.3.3 Comparison of the Proposed System with Existing Work 79
3.4 Conclusion 79

4 Deployment of Proposed Sanskrit-Hindi Machine Translation System on Cloud 85
4.1 Sanskrit-Hindi Hybrid Machine Translation System(SHH-MTS) as a Service 86

4.1.1 Characteristics of Sanskrit-Hindi Machine Translation System . . . 86
4.2 Deployment of SHH-MTS on Cloud . . . 87
4.3 Experimental Details . . . 90
4.4 Validation of Results for the Proposed System on Cloud . . . 92
4.4.1 Comparative Analysis of the Proposed Work with Earlier Research Work on Cloud . . . 92
4.4.2 Performance Analysis on Deployment of Rule-based MT, Neural MT and Hybrid MT on Cloud . . . 94
4.5 Conclusion . . . 95

5 Case Study on Proposed Machine Translation System 99
5.1 Proposed Error Taxonomy . . . 100
5.1.1 Orthography . . . 100
5.1.2 Morphology . . . 102
5.1.3 Lexical . . . 103
5.1.4 Syntax . . . 103
5.1.5 Pragmatics . . . 104
5.2 A Case Study of Proposed Error Taxonomy on Sanskrit to Hindi Language 105
5.2.1 Prerequisite . . . 106
5.2.2 Error Identification and Classification . . . 106
5.2.3 Features of the Proposed Teaching-Learning Framework . . . 106
5.2.4 Implementation . . . 108
5.3 Conclusion . . . 109

6 Conclusions and Directions for Future Research 111
6.1 Conclusions . . . 112
6.2 Directions for Future Research . . . 112

Bibliography 115

Appendix 138
A.1 Performance Evaluation . . . 138
A.1.1 Manual Performance Evaluation . . . 138
A.1.2 Automatic error evaluation . . . 139

A.2 Web interface for *Sanskrit- Hindi Hybrid Machine Translation System* . . . 141

7 List of Publications **147**

List of Papers Published **147**

List of Figures

1.1 NLP 2
1.2 Natural Language Processing involving Translation 2
1.3 Natural Language Processing Applications 3
1.4 Classification of Translation Modelling 4
1.5 Three Service Models of Cloud Computing 12
1.6 Deployment Models of Cloud Computing 12
1.7 Need for Cloud Computing in Machine Translation Systems 14

2.1 Classification of Translation Modelling 24
2.2 Rule-based Modelling for Different Languages 27
2.3 Statistical Modelling for Different Languages 29
2.4 Neural Modelling for Different Languages 31
2.5 Hybrid Modelling for Different Languages 32
2.6 Overall Percentage of Various Modelling Techniques 41
2.7 Modelling Techniques based on the Year of their Development 42
2.8 MTS Developed for Languages 43
2.9 Different Modelling Techniques for Sanskrit Language 47
2.10 Translation Systems for Processing Sanskrit 48

3.1 Deep Neural Network Architecture 60
3.2 Flow Diagram of Neural Model 70
3.3 Epochs for Training and Test Set 73
3.4 Sentence Length Affecting (a) Updates,(b) Epochs, and (c) Time 74
3.5 (a) BLEU Varies with Beam Sizes (b) Development Probability Varies Depending on Sentence Length (c) Training Probability Varies Depending on Sentence Length. 74

3.6 RBMT, Neural and Hybrid Models Across their (a) BLEU (b) METEOR . 76
3.7 RBMT, Neural and Hybrid Models Across their (*a*) Word Error Rate(WER) (*b*) F_measure 76
3.8 Comparison of Baseline System, i.e. RBMT for Sanskrit-Hindi[212] Corresponding to the Proposed System on Various Evaluation Measures 78
3.9 Human Evaluation of the Proposed System with Sahit 83
3.10 Comparison of Overall Error Rate and Accuracy (*a*) SHH-MTS and Existing Systems (*b*) [190], and (*c*) [247] 84

4.1 Flow Chart of the Proposed System 87
4.2 Architecture for the Proposed System 88
4.3 AWS Infrastructure used for Deploying SHH-MTS as a Service 89
4.4 SHH-MTS as a Service 92
4.5 Deployment and Usage of Cloud Infrastructure [206] 94
4.6 Deployment and Usage of Cloud Infrastructure for SHH-MTS as a Service 94
4.7 Average Response Time pertaining to Rule Matching Probability (a) Hybrid MTS, (b) Neural-based MTS, and (c) Rule-based MTS 95
4.8 Average Response Time pertaining to Number of Matching Action Rules (a) Hybrid MTS, (b) Neural-based MTS, and (c) Rule-based MTS 95
4.9 CPU Utilization pertaining to Packet Arrival Rate (a) Hybrid MTS, (b) Neural-based MTS, and (c) Rule-based MTS 96
4.10 Cost pertaining to Resources (a) Hybrid MTS, (b) Neural-based MTS and (c) Rule-based MTS 96
4.11 Throughput (a) Hybrid MTS, (b) Neural-based MTS, and (c) Rule-based MTS 96
4.12 Time Taken with respect to Number of Virtual Machines 97
4.13 Server Load with respect to Time 97

5.1 Framework of Linguistically Motivated Error Taxonomy for Morphologically Rich Indo-European Languages 101
5.2 Proposed Taxonomy 105
5.3 Teaching-Learning Framework 108

5.4 (a) Comparison of BI-RNN Model Across Different Parameters, (b) Evaluation of Sample Sanskrit Sentences Mentioned in Table 6.2 Across Different Measures 110

List of Tables

1.1 Properties of MTS 7
1.2 Existing Deployment of Applications on Clouds 15

2.1 RBMT Systems based on their Language Pair with the Respective Accuracy 25
2.2 RBMT System Methodology Adopted with its Specific Domains and Corpus 26
2.3 SMT System for Languages with their Respective Accuracy 30
2.4 SMT Models along with Methodology and Toolkit 33
2.5 SMT Classification-based on Different Domains and Corpuses 34
2.6 NMT System based on Toolkit with its Respective Methodology 34
2.7 NMT Systems with their Language Pairs and Respective Accuracy 35
2.8 Hybrid MT based on Language with its Respective Accuracy 35
2.9 NMT Systems with their Respective Domain and Corpus 36
2.10 Hybrid MTS Modelling Technique based on its Model and Toolkit 36
2.11 Hybrid Systems with their Domain and Corpus 36
2.12 Comparison of MTS Techniques based on Various Parameters 42
2.13 Human-Engineered Systems Developed for Various Languages 44
2.14 Machine-Engineered Systems Developed for Various Languages 45
2.15 Comparison of Modelling Techniques for Indic Language Pairs 46
2.17 Comparison of Sanskrit and Hindi Language Processing Systems 51
2.18 Existing Work of Machine Translation Deployed on Cloud 52
2.19 Existing Commercial MTS:A Comparative Study 53

3.1 Parallel and Monolingual Dataset from Different Domains 71
3.2 A Glimpe of Manually Curated Bhagavad-Geeta Parallel Corpus 71
3.3 Additional Monolingual Dataset 71
3.4 Dataset Division into Training, Development and Testing 71

3.5 Model Size 72
3.6 BLEU Scores of Different Experiments Performed 75
3.7 Metric Analysis of Sanskrit to Hindi Translation 80
3.8 A Case Study of Linguistic Analysis of Sanskrit to Hindi Translation-I . . . 81
3.9 A Case Study of Linguistic Analysis of Sanskrit to Hindi Translation-II . . 82
3.10 Adequacy and Fluency of the Proposed System and the Existing Work[247] 83

4.1 Technical Specifications along with Versions 91
4.2 Throughput Results on Standalone 93
4.3 Throughput Calculation on Virtual Machines 93

5.1 Error Identification and Classification Across Different Sanskrit Corpora . 107

6.1 Evaluation Measures 140
6.2 Library Used in the Research Work 141

Dedicated to my mother

Certificate

I hereby certify that the work presented in this thesis titled, **Cloud-based Sanskrit to Hindi Machine Translation System**, in fulfilment of the requirement for the award of degree of Doctor of Philosophy, submitted to the Computer Science and Engineering Department of Thapar Institute of Engineering & Technology, Patiala, is an authentic record of my own work carried out under the supervision of Dr. Ravinder Kumar and Dr. Inderveer Chana, and refers other researchers' works which are duly listed in the reference section.

The matter presented in this thesis has not been submitted for the award of any other degree of this or any other university.

Muskaan

(Muskaan Singh)
951503012

This is to certify that the above statement made by the candidate is correct and true to the best of our knowledge.

Dr. Ravinder Kumar
Associate Professor
CSED,TIET

Dr. Inderveer Chana
Professor and Dean, Student Affairs
CSED,TIET

Acknowledgments

First of all, I express my gratitude to the almighty who paved the way for me to complete this arduous task successfully. His blessings always worked a lot in overcoming all the difficulties i ever faced during the tenure of this research work.

I would like to express my heartfelt gratitude to my worthy supervisor Dr. Ravinder Kumar, Associate Professor; and Dr. Inderveer Chana, Professor Thapar Insitute of Engineering & Technology, Patiala for being a beacon of light and continuously guiding me through thick and thin. They, indeed, are great to have all the qualities that a pupil could have asked for in a mentor. Their constant support, encouragement, immense knowledge and perception, lit up my way in the darkest times. I am eternally indebted to them for their insightful discussions and highly valuable suggestions that assisted me in shaping up this research work. The time and energy that they devoted to correct this manuscript helped me to present this thesis with the desired quality. I cannot imagine a better mentor for this journey. Their belief in me was always re-energizing and rejuvenating.

I would like to express my gratitude to our director Prof. Prakash Gopalan; Dr. Rafat Siddique, Dean of Research & Sponsored Projects and Head of Department Prof. Maninder Singh for their constant motivation and encouragement. I would also like to thank Prof. Vineet Chaitanya, Professor, IIIT Hyderabad; and Prof. Amba Kulkarni, Professor, University of Hyderabad who guided me to improve the quality of this research work. I would like to thank Dr. Prashant Singh Rana for his kind support and motivation. I extend my sincere thanks to all the members of my doctoral committee - Dr. Seema Bawa, Dr. Ajay Kumar and Dr. Alpana Agarwal for their academic support and invaluable comments. I sincerely concede their valuable feedback and constructive comments while ensuring the progress of my research work.

I wish to further extend my thanks to Ph.D. Coordinator Dr. Sushma Jain and all the members of the faculty of the Computer Science & Engineering Department, Thapar Institute of Engineering & Technology, Patiala who helped me in one way or the other to carry out my research work successfully. I am grateful to them for their co-operation and moral support. I also acknowledge the cooperation rendered to me by the office and the laboratory staff of this department. I would also like to thank all my friends and lab mates for their continuous motivation and kind help. Also, I am thankful to them for all the great times that we have shared and for making this research experience enjoyable and memorable.

This thesis would never have borne fruit without the unconditional support of my family. My mom deserves a special mention here, who inculcated moral, ethical and

religious values in me. She made me smile and realize her faith in me during the rough road of this journey. She has passed onto me a wonderful humanitarian lineage and a good foundation to face the challenges of life. I gratefully acknowledge the patience and love of my brother in every sphere of my life. I would extend my deep sense of gratitude to one and all in the family of my parents-in-law who extended their cooperation and encouragement to me. They provided me an excellent conducive environment for continuing my research.

Last but not the least, my warmest thanks to my dear husband Vivek Sheoran, whose unconditional support and continuous motivation during all these years is beyond expression in words. He inspired me in all the dimensions of life and filled credence in me to complete this journey. I will always be indebted to him for his everlasting love,care and commitment.

Muskaan Singh

Abstract

Machine Translation(MT), one of the several applications of Natural Language Processing(NLP) enables an automatic translation of sentences or documents from one language to another. It aims at reducing the language barriers of human communication belonging to different linguistic backgrounds. Language perplexity has a tremendous impact on several aspects of human subsistence, which can be mitigated with effective use of MT. It endeavours at minimising the involvement of human-being. Although machine-generated output may differs from human translation, it is easily understandable. It manifests its effectiveness by producing grammatically and semantically fluent output.

The work presented in this thesis is a modest endeavour to study in detail the extant modelling techniques of MT. It provides a chance to deeply understand the various issues and aspects of the current study. It also serves the purpose of finding the gap in the research area and avoid duplication. It serves the developers with resource's required for modelling techniques such as corpus, domains, toolkits, models, features and their evaluation measures. Sanskrit-Hindi translation has been in existence since many years but it lacks extensibility, generalizability and adaptability which have been overcome by the proposed system developed in this research work.

In this work, we have proposed and presented a hybrid MTS for translating Sanskrit to the Hindi language. The technique developed uses linguistic features from rule-based feed to train neural machine translation system. The work is novel and applicable to any low-resource language with rich morphology. It is a generic system covering various domains with minimal human intervention. The performance analysis of work conducted on automatic and linguistic measures. It has shown through results i.e., BLEU score of 61.02% of proposed and developed system outperforms earlier work for this language pair.

The proposed MTS is deployed further on the cloud to offer translation as a cloud service and improve the quality of service (QoS). It is developed on TensorFlow and deployed under the cluster of virtual machines in the Amazon Web Server (EC2). The significance of this work lies in demonstrating the management of recurrent changes in terms of corpus, domain, algorithm and rules. The accuracy, speed and response time of the MT system are quite encouraging and satisfactory. The proposed hybrid model is faster and more efficient than the existing rule-based systems. In non-rule match cases, the rule-based model does not return any output however, the proposed model has always returned the best

solution. The existing model is quite complex for long sentences, and sometimes these are practically infeasible but the proposed model is efficient in such cases also. OI98The system on cloud is evaluated for different QoS parameters like response time, server load, CPU utilization and throughput. The experimental results asserts, with the availability of elastic computing resources in the cloud environment, the job completion time irrespective of its size can be assured to be within a fixed time limit with high accuracy.

The work presented in this thesis has been validated with a case study presented at the end. It outlines the developed taxonomy of error analysis based on different linguistic levels, i.e., orthography, morphology, lexical, syntax, semantics and pragmatics. Consequently, the previous taxonomies were expanded to adapt the errors transpired in morphological rich Indo-European languages. The MTS employed for the case-study is developed as a service using linguistic analysis along with deep learning to aid the teaching and learning process. As far as direct access to Sanskrit text is concerned, it requires a good grammatical knowledge, manual access to the dictionary, knowledge of syntax and semantics which is a tough and time-consuming process. This interactive interface will assist the school as well as university students enrolled in distance education by promoting self-learning. The main aim of the proposed system is to make the scriptures and philosophical texts such as Gita, Ramayana and Upanishads, available in the Sanskrit language, accessible to the common user. It also substantially provides future research directions and aid in the human error analysis process.

Chapter 1

Introduction

India is a country of immense language diversity having more than 1.3 billion people, 30 official languages, and more than 1 million speakers. Linguistic diversity has its own advantages; it transmits not only different cultures, but also helps in finding the best solution to any problem. Thus, translating the content from one language to another is of the utmost significance. The translation use case range from government, enterprise to social.

Machine Translation(MT) is one of the key areas of Natural Language Processing (NLP) and computational linguistics. It is important for breaking the language barrier and facilitating inter-lingual communication. For a multilingual country like India, there is a big requirement for such a system. With the advent of information technology, many documents and web pages are available in a local language. So, there is a large need for a good Machine Translation System(MTS) to address all these issues to establish proper communication among the people of different backgrounds and cultures.

This chapter provides an overview of the current research work. It discusses the fundamental concepts related to machine translation and cloud computing. It further unfolds its close alliance with other underpinning technologies and introduces the various issues of this research area. Henceforth, it motivates the research, problem statement, objectives and culminates with the discernment of the contributions and organization of the rest of the thesis.

1.1 Machine Translation:An Overview

Artificial Intelligence(AI) aims at developing an intelligent system examined by humans intuitively[1]. Natural Language Processing (NLP) is one of many applications of AI. NLP is an area of research and application involving computers to understand the text in natural language. It is a multidisciplinary field involving computer scientist and computational linguistics as shown in Figure 1.1. It builds a computational model for its analysis and generation as shown in Figure 1.2. It involves technological, cognitive and linguistic motivation for developing intelligent computer systems such as machine translation, information extraction, sentiment analysis, speech recognition, text classification, etc. It involves natural language understanding and its generation as in Figure 1.3. Machine Translation(MT)is one of the key applications of NLP[2][3].

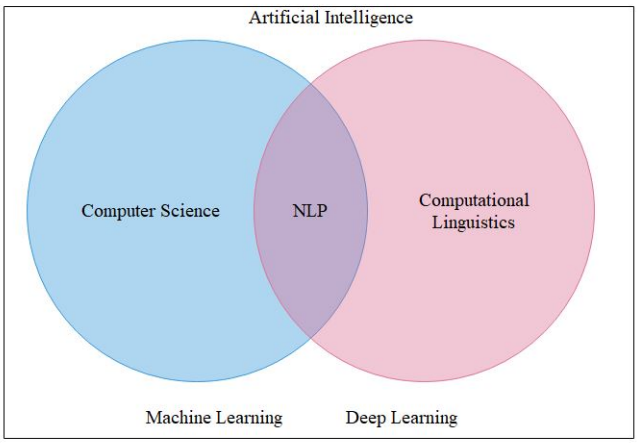


Figure 1.1: NLP

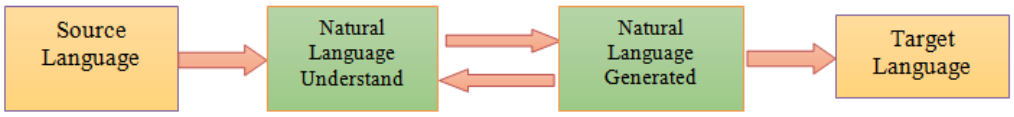


Figure 1.2: Natural Language Processing involving Translation

MT is a process of translating source language to target language using a computerised system. Human translators or editors can be involved in the process of MT, although minimal human aid is the goal of MT. The field of man-machine interaction involves the processing of natural language. Some of the technologies that contribute to the development of MT have been listed below:

- *Computational Linguistics*: It covers word formations and ordering, analysis of meaning and other communication aspects.
- *Knowledge Representation*: It is an area that deals with formalisms used in logic, frames and semantic networks.
- *Semantic Network*: It provides a linkage through relationships of concept collections.

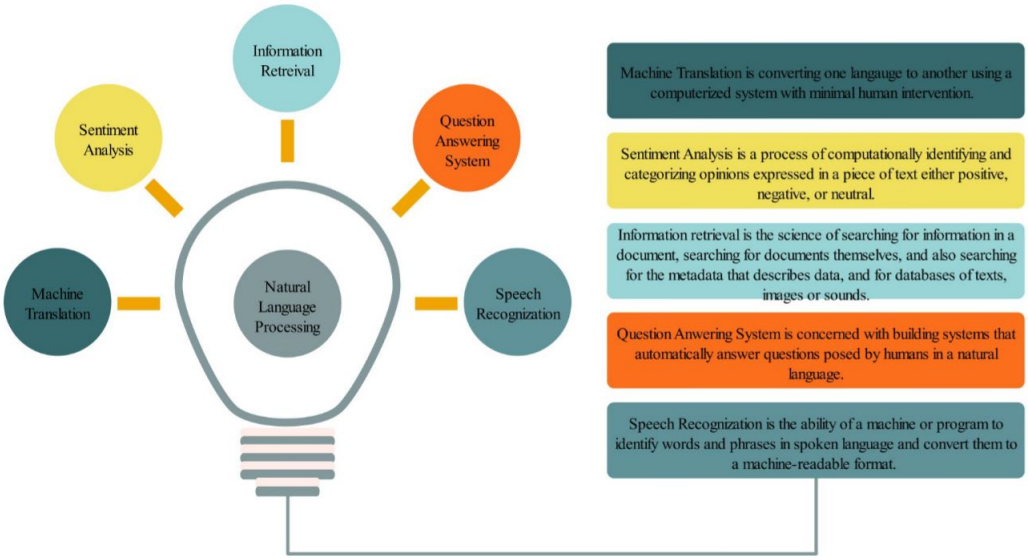


Figure 1.3: Natural Language Processing Applications

- *Machine Learning*: It is a field where the machine learns using different machine learning models for the accession of new knowledge from data.
- *Search Algorithms*: It assists in finding a solution to the problem and not getting stuck in an infinite loop.

A translation provided by the Machine Translation System(MTS) involves both syntactic and semantic aspects of language to be covered to provide the correct version of translation.

1.1.1 Machine Translation Approaches

The classical translation approaches delve the deep insight of linguistic knowledge of source language and cognitively translate to target language word by word. The approaches for MT are categorized predominantly into Rule-based, i.e., based on linguistic handcrafted rules[4] involving transfer-based mechanism [5] and interlingua mechanism [6], corpus-based approach that is entirely based on corpus, i.e., statistical phrase-based[7], and neural-based[8]. Even example-based[9] and Knowledge-based[10] are less used nowadays. MTS is built using various modelling techniques. It depends on the number of criteria such as resource availability (parallel corpus, monolingual corpus, human resource, lexical resources and technical resources), the background of the developer (linguists prefer rule-based technique; translators prefer example-based technique; computer-scientists prefer interlingua-based technique; statisticians prefer statistical-based technique; and mathematicians prefer neural-based technique), and the goal or purpose of MTS (assimilation, dissemination, one or more language pair, general or specific domain). MTS has

been classified based on engineering involved for developing it either Human or Machine. Human-engineered modelling techniques are Dictionary or direct modelling technique and Rule-based modelling technique, whereas Machine-engineered modelling technique is Corpus-based, and Hybrid-based modelling technique as shown in Figure 1.4.

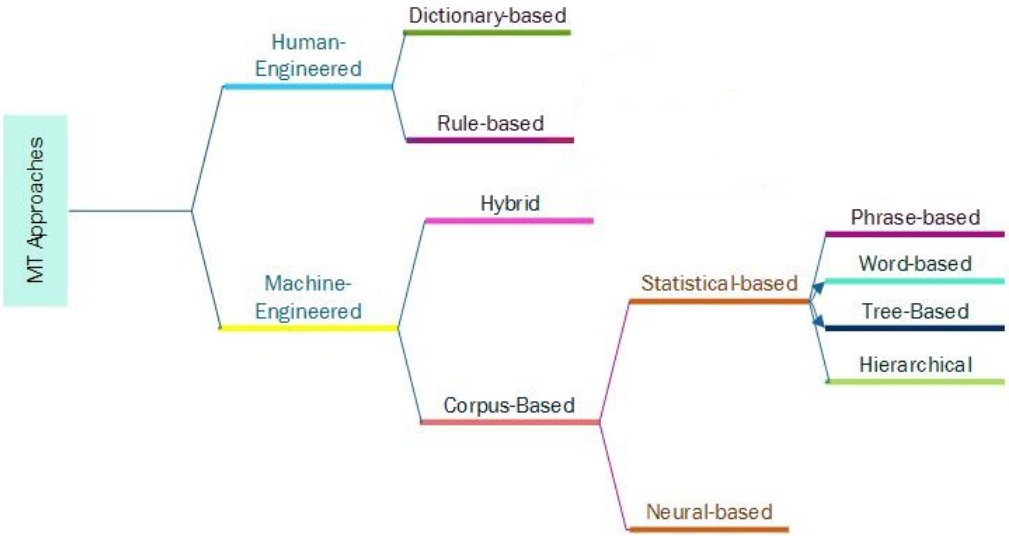


Figure 1.4: Classification of Translation Modelling

1.1.1.1 Human-engineered Translation Techniques

These modelling techniques involve more human intervention as all the modules of these systems require human insights. These can be classified as Direct and Rule-based modelling technique.

- Direct MTS involves word to word translation using bilingual dictionaries. This is one of the simplest and easiest ways to implement MTS. Syntactic structure and semantic relations are not concerned with direct MTS to involve word to word translation using bilingual dictionaries. The following steps may be followed[11]:
  - *Root word identification*: The source document comprises of various words, of which root word needs to be identified by removing the suffixes.
  - *Dictionary lookup*: To fetch target document words, a look-up of the dictionary is performed.
  - *Post-editing*: Words are re-ordered to achieve the final translated target document.
- Rule-based MTS(RBMTS): The entire process (morphological analysis, syntax analysis, semantic analysis, and Target language generation) of rule-based MTS depends on rule specification. Therefore, it is termed as rule-based MTS. Rule-based modelling technique converts parsing of the sentence to intermediate structure, which is further converted into target[12]. Rule-based MTS is further divided into two types as below:

– Transfer-based MTS(TBMTS): As stated earlier in the rule-based modelling technique, the parsing of the input sentence is an important step. In the transfer-based system, the parsed input sentence is transferred to the target language parse tree. The target-language text is then generated from the destination language parse tree. There are three different steps followed by the transfer-based modelling technique, i.e., analysis, transfer, and generation. This type of system is based only on syntactic features[5].
– Interlingua-based MTS: It is another kind of rule-based system. In this, the source language text is converted into an intermediate structure. Through the interlingua, any target language can be generated with minimal efforts. Two steps are followed to convert the source language to target language, i.e., analysis and synthesis. This type of MTS is independent of the language pairs. The intermediate representation that applies to all kinds of MTS is difficult[6].

1.1.1.2 Machine-engineered Translation

These modelling techniques involve more Machine intervention as all the modules of the system require machine processing. It builds a translation model learned from the corpus (monolingual and bilingual) as displayed in Figure **??**. It is a process of converting source language to an intermediate representation. The intermediate representation is then converted into the target language. Different types of machine-engineered MTS are:

- Corpus-based MTS: It depends entirely on the corpus (bilingual, multi-text or parallel corpus). MTS developes automatically with the help of corpus. In comparison to the rule-based modelling technique, it requires more efforts as it is fully automatic. These are further categorized into two types as below:

  – Statistical-based MTS (SBMTS) This statistical model is used for developing MTS. Its parameters are derived from parallel corpus by applying training and machine learning models which generates a statistical table. The last step in building a statistical MTS is decoding. In this, the best MTS is used for source-language text. The statistical modelling technique is further divided into these types: word-based, phrase-based, syntax-based, and hierarchical phrase-based[13].
  – Neural-based MTS: The neural modelling technique of MTS is gaining a significant interest of researchers. It is also known as an Encoder-Decoder approach. It consists of two components, viz. source sentence X and target sentence Y which are decoded. The goal is to maximize the conditional probability of paired sentences of the parallel training corpus. This forms a parameterized model. The conditional probability of Y gives a source sentence X, i.e., arg MAX(Y) P(Y/X). A conditional distribution is learned by the model for source sentence and the target sentence is generated. Searching for the sentence maximizes conditional probability[14][15][16].

- Hybrid-based MTS (HBMTS): These kinds of MTS combine both statistical and rule-based modelling techniques. This method improves the efficiency of MTS as the limitation of one technique is overcome by the second technique. There are two modelling techniques that can be followed by the hybrid system. In the first technique, the rule-based system output is fed to statistical-based modelling technique[17]. In the other techniques, the statistical-based output is supplied to the rule-based modelling technique [18].

1.1.2 Properties of Machine Translation System

There are various properties of MT such as Quick translation, enhanced timeline, confidentiality, cost, universality, consistency and online translation and translation of web page content. The description of these properties covered in detail in Table 1.1.

1.1.3 Applications of Machine Translation

MT has widespread commercial, military, and political applications. For example, increasingly, the web is accessed by non-English speakers, reading non-English pages. The ability to find relevant information clearly should not be restricted by our language-speaking capabilities. Furthermore, we may not have sufficient linguists in any language of interest to cope with the sheer volume of documents that we would like translated. MT poses several interesting machine learning challenges: data sets are typically large, as are the associated models; the training material used is often noisy and plagued with sparse statistics; the search space of possible translations is sufficiently large that exhaustive search is not possible. Some of the translation use cases have been listed below:

- Government
  - Administrative requirements
  - Education
  - Security
- Enterprise
  - Product manuals
  - Customer support
- Social
  - Travel (signboards, food)
  - Entertainment (books, movies, videos)

Table 1.1: Properties of MTS

Properties	Description
Quick Translation	The use of MTS enables us to save our time while translating large texts.
Enhanced Timeline	A key benefit of MT is speed. It is significantly faster than human translators. It is estimated that an average human translator can translate around 2000 words a day, while multiple translators can be assigned any given project to increase that output. It still pales in comparison to MT. It can generate thousands of words every minute.
Confidentiality	Many people use MTS to translate their private emails because no one would agree to give his private correspondence to a translator whom he doesn't know, or no one would entrust financial documents to other people.
Cost	The cost of MT is also significantly lesser than that of human translation. There are extremely advanced MT platforms now that are accessible for free. Google translate for one is becoming more accurate each year; users can also enter an unlimited amount of translation content for free.
Universality	Usually, a professional translator becomes specialized in a defined field, but the MTS can translate any text irrespective of the area. For the translation of special terminology, we need to just switch to a corresponding setting.
Consistency	MT allows systems to memorize key terms and phrases that are used within a given industry. This leads to increased consistency across the entire text. If human translators are used for translating a text, their translations may change slightly depending on several factors, whereas an MT will not.
Online translation and translation of web page content	The advantage of online translation services is obvious. Online translation services are at hand, and one can translate information quickly with this service. Furthermore, one can translate any web page content and query of the search engine by the use of MTS.

1.1.4 Need for Machine Translation Systems

The tremendous increase in industrial growth over the past decades has a significant impact on the global MT market. It enables content to be available in all regional languages across the globe. Computational activities have become mainstream nowadays. As the internet opens up the wider multilingual and global community, research and development in MT continue to grow at a rapid rate. MT system plays a role in small, medium and

large organisations. Some deal with only a specific, but others provide services in multiple domains. Various fields are requiring domain-specific services such as government, software and technology, healthcare, legal, military and defence, e-commerce, finance and many more. The other online MTS, i.e., commercial deal with instant translation where the source language is converted into target language for text, image and audio data. These MT systems are generic, light-weight, cloud-based and have high accuracy. These offer translations like text-to-text, speech to speech, text to speech, speech to text and image to text. These commercial systems are not economical. Thus, it necessitates a need for economic MT system providing translation services.The Sanskrit being one of the ancient and termed to be the father of all languages of India is diminishing its eminence. We envisage Sanskrit to Hindi machine translation can provide a platform to study the detailed analysis of any Sanskrit text to Hindi. Even the ancient literature such as Vedas, Upanishads, Ramayana, Mahabharta etc of Sanskrit which doesn't have an existing Hindi translation could be made accessible.

1.1.5 Characteristics for Sanskrit Language Processing

According to the Census of India 2018, Sanskrit is the mother tongue of 24,821 people and Hindi of 52,83,47,193 people, i.e., 43% of total languages in India[19]. The main reason to choose Sanskrit for translation is the richness of its scientific literature with extensiveness and comprehensive analysis, structured approach and traditional grammar[20]. There are numerous characteristics of this language. Some of these are listed As hereunder:

- It is also considered as the 'donor' of almost all the Indian languages. Most of the languages have been derived from Sanskrit, either partially or fully.
- The vast reserves in this language could be converted into other languages[21]. As this language comprises of rich literature, Ayurveda, Vedas can be produced in other languages also to improve their accessibility.
- It holds a rich grammar confined by Panini near 2500 years ago formulating 3,949 rules, which extended later on[2].
- It has the strongest and non-ambiguous grammar[22]. Many people have attempted to write a grammar for Sanskrit language using the Paninian framework and used it to develop translation system[23].
- Panini focused on decoding the information contained in the language string of particularly given input language by Karaka (syntax-semantics) relations and non thematic roles[24]. The framework also highlights the importance of case markers, postpositions, and word-order. The central element of a semantic model in Paninian framework is that every verbal root (dhaatu) denotes an action consisting of an activity (or vyaapara),

and a result (or phala). The result is a state reached after completion of an action. An activity consists of steps that are performed by different participants or Karakas involved in the action.

- The concept of Karaka relation is a central theme to Paninian grammar. These Karaka relations refer to syntactic-semantic relations, and on a surface level, it highlights syntactic information and also captures semantic information at a deeper level.
- The Sanskrit grammar is termed as 'Father of Informatics' as it builds a relationship between speech and utterance of speaker and meaning derived by the listener[25].
- The primary objective of Sanskrit Paninian grammar is to form a theory of human natural language communication.
- Sanskrit and Hindi belong to the same Indo-Aryan family[26]. They both have structural and lexical similarity as Hindi inherits from Sanskrit.
- Sanskrit has the rich and structured grammar in the form of Panini Astadhayayi, whereas in Hindi such parallel grammar does not exist. Therefore, it becomes difficult to map the divergence between these two languages.
- The non-existence of parallel grammar leads to exceptional cases which uncover linguistic generalisations such as Vibhakti in Hindi. The cases where Vibhakti in Sanskrit and Hindi diverges [26] are optional, exceptional, differential, alternative, non-Karaka, verb, and complex-predicate divergence.

Despite these features, choosing Sanskrit as a source language is difficult on using both rule-based as well as neural-based approach. In the rule-based approach, parsing fails due to its synthetic nature in which single word can run up to 32 pages. Whereas in the case of training, the translation system using NMT approach leads to the high occurrence of Out-of-vocabulary words. These words are morphologically rich words, carrying multiple meanings according to the context. The work presented in this thesis clearly outlines these challenges and overcomes them by providing a sound translation.

1.2 Cloud Computing: An Overview

Cloud computing has emerged as a new paradigm of distributed computing that has migrated the computing and the data from desktops, into data centres [27]. It is an internet-based approach for empowering suitable, on-demand network access to a communal pool of computing resources like networks, servers, storage, applications, services etc. These resources can be provisioned quickly and de-provisioned with nominal management or interaction of service provider which, in turn, stimulates accessibility [28] [29] [30] [31] [32]. To rapidly and effortlessly adjust the capacity of data centres, the scalable IT resources,

as well as the underlying infrastructure, are offered on pay-per-use basis over the Internet, thereby, accommodating the changes in demand and helping any organization to avoid the capital costs of software and hardware [33] [34] [35]. The National Institute of Standard and Technology(NIST) has defined, "cloud computing as a model that facilitates expedient and dynamic access to a large pool of computing resources that can be shared, dynamically allocated, and discharged without much managerial involvements or service provider assistance" [36] [37] [38].

Cloud computing architecture establishes itself on certain key characteristics, a few of which are overlapping with Grid, Utility, Cluster and distributed computing [39] [31] [32]. These characteristics enlisted as follows:

- Multi-tenancy: The services owned by multiple cloud providers in a cloud environment are co-located in a single data centre, which is called multi-tenancy. The layered architecture of the cloud computing paradigm naturally divides the responsibilities, where each layer focuses only on the respective objectives of that layer. Multi-tenancy along with its benefits also brings the difficulties in managing and coordinating interactions among various involved stakeholders.

- Shared Resource Pooling: The cloud infrastructure provider, generally, offers a big pool of computing resources such that they can dynamically be assigned and re-assigned to the multiple cloud service consumers. This capability of dynamically assigning the resources enhances flexibility in managing the resource utilities and operating costs of the cloud service providers.

- Geographic Distribution & Ubiquitous Network Access: The services of the cloud are accessible through the Internet, and hence, use the Internet as a network for delivering the services to cloud users. Moreover, to offer improved network performance, most of the cloud data centres are distributed geographically. Thereby cloud vendors can benefit from this geo-diverseness, and thus, realize the higher utility of services.

- Service-Oriented: The cloud paradigm employs a service-driven operating model, and therefore, strongly emphasizes on the service management. Each service vendor offers IaaS, PaaS & SaaS cloud services by the negotiated Service Level Agreement (SLA) with the clients. Thus, every cloud service provider ensures integrity and service provisioning with SLA assurance.

- Dynamic Resource Provisioning: This is a key characteristic of cloud computing, where computational capability can be acquired and released dynamically. Contrarily, the conventional computing involved resource provisioning as per the extremum demand. The resource allocation carried out dynamically facilitates service deliverers in obtaining the resources as per the required demand. This aids in lowering operating costs significantly.

- Self-Organizing: The resource allocation and de-allocation are done on a demand basis, empowering the service providers to manage their resource utilization as per their needs. The programmed resource management feature enables the service vendors to quickly react to the prompt changes in the service demand as surge computing.
- Utility-based Pricing: A significant characteristic of the cloud is that it offers a pay-per-use pricing model. Although the pricing schemes vary according to the requested services, the utility pricing models considerably lower the operating costs because the users are charged on a per-user basis. This feature of cloud computing introduces complexities in controlling the operating cost.

After giving an overview of cloud computing and its key characteristics, the following subsections deal with the various service models, deployment models, and the components of cloud computing respectively.

1.2.1 Cloud Service Models

Cloud computing services have been classified into three categories as shown in Figure 1.5.The three types of cloud service models offered by cloud computing are described below [31] [32] [40]:

Software-as-a-Service (SaaS): It is a multi-tenant platform for offering applications on the Internet, boxed as a distinguished service for users to access, for example, Google Docs, Facebook, etc.

Platform-as-a-Service (PaaS): It is a framework that provides a specific computing platform where applications and services are developed, i.e. it is a model that offers building, testing, deployment, and hosting environments for applications created by users. e.g. Microsoft Azure and Google App Engine.

Infrastructure-as-a-Service (IaaS): It is a framework that provides entire computing resources through a service. Users can hire or purchase required computing resources for use without operating or managing infrastructure [41]. Examples include Amazon EC2, Eucalyptus, and Nimbus.

The further classified cloud services can be Hardware-as-a-Service (HaaS) and Data-as-a-Service (DaaS) [42] [43] [44].

1.2.2 Cloud Deployment Models

Cloud computing is mainly classified (as shown in Figure 1.6) based on variation in the physical location and distribution, i.e., the cloud can be implemented through various types of deployment models as described below [31] [32]:

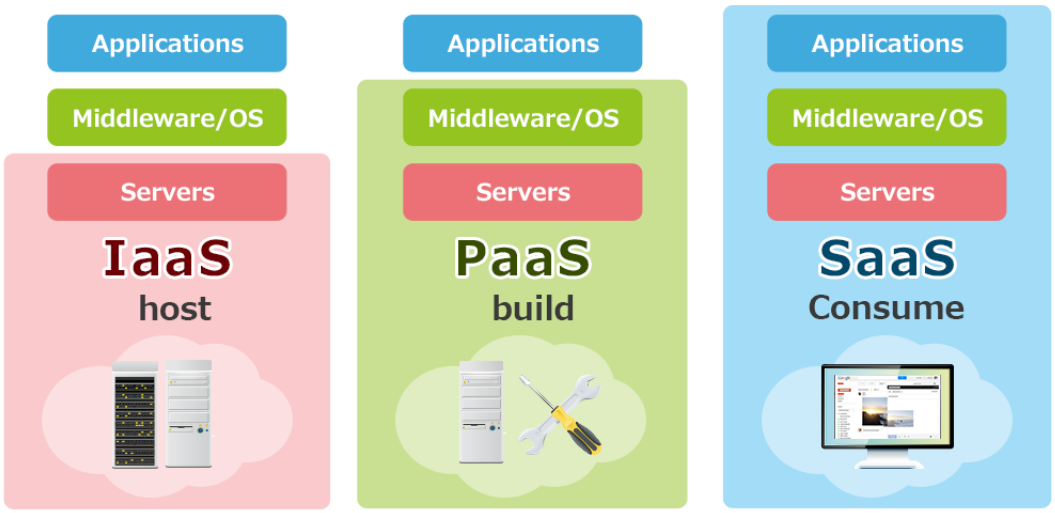


Figure 1.5: Three Service Models of Cloud Computing

- Public Cloud: It can be defined as "a cloud that is made available in a pay-as-you-go manner to the general public" [27]. These clouds are accessed by the common people and are maintained by cloud service providers like Microsoft, Google, Amazon, etc. by charging the users according to their usage. These are applicable when the service provider wants users to access all the resources over a network.

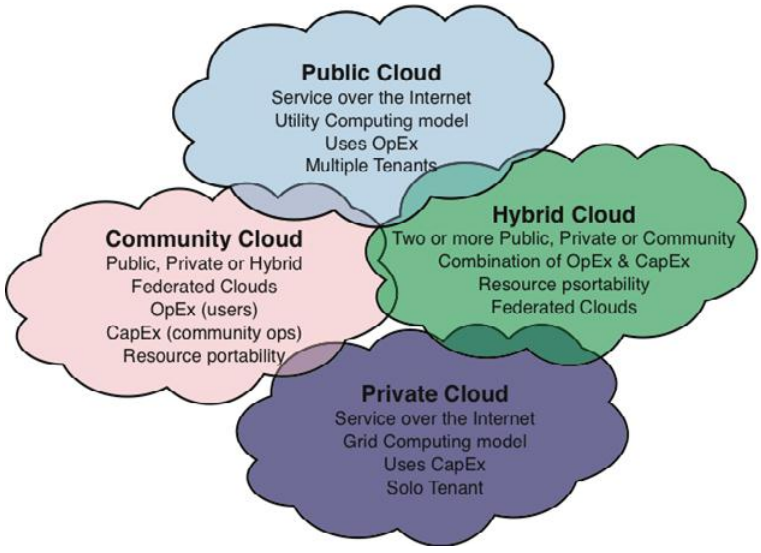


Figure 1.6: Deployment Models of Cloud Computing

1.2.3 Cloud Computing Components

Cloud computing constitutes a virtualized pool of infrastructure resources with applications and services that can be used directly through a self-service portal. Cloud computing architecture consists of the following components [31] [32]:

- Clients: A client interfaces with a cloud through a pre-assigned, subtle layer of abstraction. This layer carries out communication between the user requests and the displayed data which is returned simply and intuitively for the user, e.g., Web Browser or a thin client application.

- Data Centers: It is a collection of the different type of servers where the application is deployed and is available to various clients upon subscribing. It is a repository, either virtual or physical, used for the storage, management and distribution of the data as well as the information organized in the context of a particular topic.

- Distributed Servers: These are the servers which are present in geographically distributed locations, thereby, enabling the service providers more flexibility in terms of operations as well as security. The implementation of the distributed servers requires various techniques for maintaining the central control as well as the coordination of the information related to the configuration of the involved servers.

- Cloud Network: A network is a link connecting the wide user-base with the available cloud service providers. Undoubtedly, the Internet is the most straight-forward and popular choice to act as a cloud network. Although, advanced network services, such as compression, encryption, decryption during data at rest and data at transit will be beneficial for both the cloud user as well as the cloud service provider.

- Cloud API: A cloud Application Programming Interface (API) contains a set of programmable instructions and techniques that provides an abstraction over a cloud provider. It encompasses a customized or a discrete provider call which may be used to increase the amount of control over the implementation of cloud. API calls are used to build applications for communicating with and accessing cloud services.

After reviewing the fundamentals of cloud computing such as its key characteristics, service & deployment models and its components, the next section unfolds its need in machine translation as represented in Figure 1.7.

1.2.4 Need of Machine Translation Deployment on Cloud

Cloud computing is a model which enables ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (networks, servers, storage, applications and services) that can be rapidly provisioned and released with minimal management efforts or service provider interaction[45]. Clouds are classified based on service type or deployment type[46]. It consist of three service models SaaS, PaaS and IaaS.[47][48]. Numerous applications are being deployed on cloud nowadays as listed in Table 1.2. It includes MT application offered through the cloud and its outcome achieved by reducing the deployment time and auto-scaling of applications. Many factors gov-

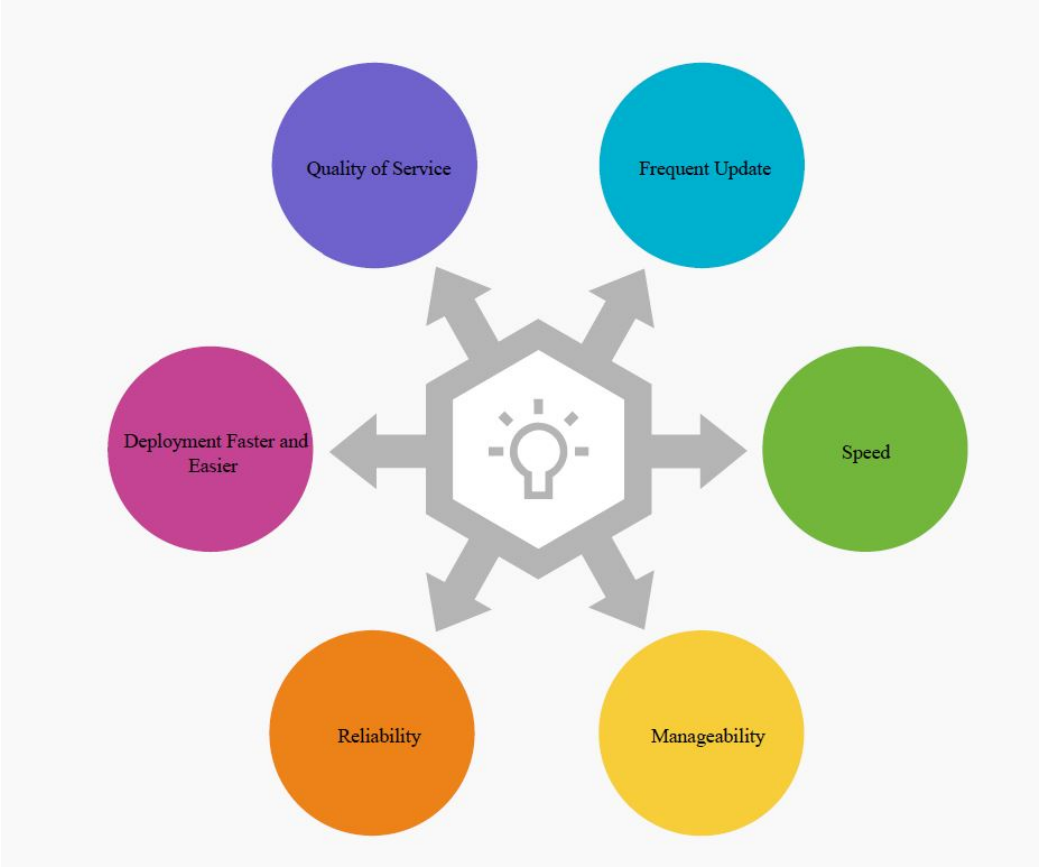


Figure 1.7: Need for Cloud Computing in Machine Translation Systems

ern the need for MTS deployment on the cloud such as deployment faster and easier, handling of frequent updates, manageability, speed, reliability, Quality of Service(QoS), enhancement, rapid provisioning, elasticity and resilience. An experiment of deploying MT system (Hindi to Punjabi) on the cloud was conducted. It took a book of 70 pages to translate on stand-alone machine 71 minutes whereas on the cloud it turned down to 5 minutes,[49]. Hence, deployment of MT systems on the cloud is a viable option for future users.

As the workload increases, change in the deploying environment is needed for such complex compute-intensive applications. It improves throughput, response time and reduces deployment time. These systems are offered by both academic and commercial organisations and have been providing services to the end-users.

1.3 Problem Statement

India is a country of enormous language diversity, with more than 1.3 billion people who have 22 official languages and more than 12 scripts; hence there is a need to provide a translation of the content from one language to another language. According to the census of India 2018, Sanskrit is the mother tongue of 24,821 and Hindi of 52,83,47,193

Table 1.2: Existing Deployment of Applications on Clouds

Domain	**Specific Application**	**Outcome achieved**
Natural language processing	Machine translation system[49]	→ Deployment time reduced. → Provision for auto scaling of application.
Cloud-based application	3-D Application[50]	→ Virtual graphics processing optimised. → Cloud-based framework for 3-D virtual appliance.
Geo-spatial Research	Model processing[51]	→ It takes task of model processing field. → open source software is used.
Operating system	File system related development[52]	→ Virtual machine can be migrated. → Cache for unified buffering. → File system is portable.
Educational field	Teaching operating system[53]	→ Helpful to students for assignment as it can be deployed on their computers. → Can be practised without the facility in labs, and gives kernel level project experience.
Server dependencies	Legacy distributed system[54]	→ Generates graphs with dependency representation. → Optimum deployment on server framework.

i.e. 43% of total languages in India; hence we choose this language pair to translate from Sanskrit to Hindi[19] We formulate our problem statement pointwise.

- Firstly, machine translation systems are required for people to communicate with the enormous language diversity in India. There are various contents on the web, which are in Sanskrit language such as blogs, stories, poems, literature, news etc., which are dynamic in nature and can be easily translated to their regional language using MT.

- To the best of our knowledge, not all the life-transforming stories (epics, vedas, upanishads, ramayana, mahabharata) are available in Hindi with the detailed analysis of the source text.

- Detailed analysis can be done with the linguistic tools of Sanskrit to capture its morphological rich feature; hence we term it as a complex system as it consist of pipeline architecture. Deploying such a system on standalone itself requires NLP prerequisites, memory and much time for usage.

Even this system requires frequent updates in terms of algorithm, corpus, dictionary, rules, linguistic tools which is cumbersome and time consuming. To overcome such a problem, the system can be deployed for experimental purposes on cloud to provide quality of service and better performance.

1.4 Research Motivation

The research motivation of this work, was to serve the community with in-depth analysis of Sanskrit to Hindi language translation. We have elaborated the motivation behind the research work pointwise.

- We envisage to serve as a platform for learning and accessing Sanskrit language for in-depth analysis. We empirically believe that Sanskrit being, enriched with scientific, extensive and comprehensively analysed structured grammar is losing its eminence in today's fast paced world. This would call for a hybrid sanskrit-hindi machine translation system which combines the best of both the rule-based and neural-based approaches. However, to apply a neural-based approach to build such a system requires an adequate amount of parallel corpora which is not available to the best of our knowledge.

- We also believe a Sanskrit-Hindi machine translation system, can provide assistance to students learning through distance education and virtual interactions going mainstream to the world adapting to the new normal in the Covid-19 pandemic. Even various contents on the web, which are in Sanskrit language such as blogs, stories, poems, literature, news etc., which are dynamic in nature and can be easily translated to their regional language using MT.

- The deployment of such a system requires prerequisite knowledge of NLP, which we have dealt in this work by deploying the system-as-a service on cloud. There is also a much common problem of frequent updates in terms of algorithms, linguistic tools, rules, domain adaptation, corpora etc. to improve the quality of translation. Deploying such a system on cloud will provide efficient, scalable, faster, manageable and reliable system services.

1.5 Objectives of the Proposed Work

The main objectives of the proposed work are as follows:

1. To analyse and develop lexical resources for Sanskrit to Hindi machine translation systems.

2. To propose and develop a Sanskrit to Hindi machine translation system using a hybrid approach by incorporating various lexical resources required for the generation of the target language.

3. To test the performance of the proposed machine translation system in a generic and cloud environment for offering QoS (Scalability, manageability, performance, etc.) to end-user.

To attain the first objective, a comprehensive investigation has been conducted to study the various existing lexical resources; and their relative pros and cons have also been identified. The existing survey provides that the lexical resources present for the Sanskrit to Hindi language pair are quite less. Thus the current work is a modest attempt to manually develop a parallel corpus of Sanskrit to Hindi for Bhagavad Geeta. Based on the findings of the literature survey, the research problem has been defined.

To achieve the second objective, a Sanskrit-Hindi Hybrid Machine Translation System (SHH-MTS) model has been proposed and developed. The developed technique uses linguistic features from rule-based feed to train neural machine translation system. The work is novel and applicable to any low-resource language with rich morphology. It is a generic system covering various domains with minimal human intervention. The performance analysis of work has been performed on automatic and linguistic measures. The results achieved exhibit that the proposed and developed approach outperforms the earlier works undertaken in the field.

To accomplish the last objective, the proposed machine translation system is deployed on the cloud to offer translation as a cloud service and improve the quality of service (QoS). It was developed on TensorFlow and deployed under the cluster of virtual machines in the Amazon Web Server (EC2). The significance of this objective lies in demonstrating the management of recurrent changes in terms of corpus, domain, algorithm and rules. Further, the work also compares the MTS as deployed on a stand-alone machine and on the cloud for different QoS parameters like response time, server load, CPU utilization and throughput. The experimental results assert that in the translation task, with the availability of elastic computing resources in the cloud environment, the job completion time irrespective of its size can be assured to be within a fixed time limit with high accuracy. It also assists in the deployment of MT application without requiring knowledge of NLP, making it convenient for a common user to utilise such a complex application.

1.6 Thesis Contribution

This research contributes significantly in the following ways:

- An extensive literature survey of the work carried out in the area of machine transla-

tion was undertaken to identify the gaps in the research area for devising an effective MT technique. The review of different modelling techniques with the perspective of resources has been covered in this work. It serves the developers with resources required for modelling techniques such as corpus, domains, toolkits, models, features and their evaluation measures. A comparison of research works on different Indic language pairs based on modelling techniques was performed. The survey of modelling techniques involves Indian languages such as Punjabi, Bengali, Marathi, Telugu, Tamil, Assamese, Urdu, Malayalam, Gujarati, Sanskrit, Kannada, Dogri, Sinhala, and Devnagari. The conclusion drawn after the synthesis that works for the Sanskrit language was minimal despite being such a rich literature. Therefore, this work contributes to the research of MTS for processing the Sanskrit language. The critical technical and linguistic challenges that are likely to be faced in building MTS for the Sanskrit language have been reported in depth.

- Parallel corpus available from different sources gathered while the rest of the data manually curated for building a Neural Model using Encoder-Decoder architecture with an attention mechanism.

- A NMT system was designed trained and tested with different models, activation functions, training data, epochs, sentence lengths to yield better accuracy of the proposed work. The linguistic tools output was merged from the classical rule-based approach as features embedding matrices in NMT to test whether it guided to get an accurate translation of words as these may have different meanings in some other context. It also verifies whether the proposed system helped to reduce the data sparseness and performed meaningful tokenisation.

- Performance testing carried out to compare rule-based, neural and hybrid systems on four measures, i.e. BLEU, F-measure, METEOR, Word-Error-Rate. Various Sanskrit to Hindi MT sentences evaluated corresponding to reference translation based on BLEU score, Precision, Recall, F-Measure, WER, F-mean, Penalty, and Meteor. Human-based evaluation was performed by taking into consideration the grammatical category. As many as 15 cases were included to form the grammatical categories and tested the system translation corresponding to them. The existing system for Sanskrit-Hindi translation was compared with the proposed work.

- A web-interface was developed for better user-interface. The web-interface was deployed on AWS cloud with EC2 which eased time, knowledge and complications. It even becomes challenging for a common user to utilise such a complex application. It elevated the MTS performance by managing recurrent changes in terms of either corpus, domain, algorithm and rules.

- This thesis contributes a case study of the proposed error taxonomy was also made to

test the performance of the proposed machine translation system to achieve an effective translation from Sanskrit to Hindi.

1.7 Thesis Organization

After providing an overview of the research problem and the thesis in chapter 1, the rest of the thesis is organized as:

Chapter 2 *Literature Survey*: This chapter discusses the various MT modelling techniques followed by a detailed survey of the existing MTS based on these techniques. Then it highlights the challenges in the field of translation. It serves the developers with resources required for modelling techniques such as corpus, domains, toolkits, models, features and their evaluation measures. It makes a comparison of research work on different Indic language pairs based on modelling techniques. This chapter reviews the frequently used approaches of RBMT and SMT. However, the accuracy and efficiency of NMT and HBMT techniques were better. It also provides future research directions in the field of MT for work processing Sanskrit language. It carries out a discussion on the research work for the Sanskrit language. Then, it carries the chapter further by contributing to open issues, technical and linguistic challenges along with future research directions in the field of MT for processing the Sanskrit language.Chapter 2 is partially derived from:

- Singh M, Kumar R, Chana I. Machine Translation Systems for Indian Languages: Review of Modeling Techniques, Challenges, Open Issues and Future Research Directions. Archives of Computational Methods in Engineering. 2020 Jun 17:1-29.

- Muskaan Singh, Ravinder Kumar, and Inderveer Chana, "Corpus based Machine Translation System with Deep Neural Network for Sanskrit to Hindi Translation". Procedia Computer Science 167 (2020): 2534-2544.

- Muskaan Singh, Ravinder Kumar, and Inderveer Chana, "Encoding-Decoding Methods for Neural Machine Translation". 2nd International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT),Kerala, India, Vol. 1. IEEE, 2019.

Chapter 3 *Proposed Sanskrit-Hindi Hybrid Machine Translation System(SHH-MTS)*: The proposed Sanskrit-Hindi hybrid machine translation system involves three different translation models (Rule-based, Neural-based and hybrid) that have been produced and analyzed for Sanskrit-Hindi. MT is a challenging job; hence, the model becomes compounded and time-consuming. Earlier works lack extensibility, generalizability and adaptability which have been overcome by the proposed system. The work developed and presented here is novel and can be applied to any low-resource language pair with minimum linguistic knowledge. The work extracts features from the linguistic rules and further

passes these features to train the recurrent neural network. Performance evaluation made on the automatic and human measure has resulted in improving the performance of the hybrid systems. In the non-rule match case, the rule-based model provides no output. The proposed model has always given the best solution. The existing models are complex for long sentences and are practically infeasible, but the proposed model is efficient in such cases also. The future work would involve multiple linguistic languages converted into the single target language and multi-lingual platform for the same purpose. Chapter 3 has been derived from:

- Singh, M., Kumar, R. & Chana, I. Improving neural machine translation for low-resource Indian languages using rule-based feature extraction.Neural Computing & Application(2020). https://doi.org/10.1007/s00521-020-04990-9.
- Muskaan Singh, Ravinder Kumar, and Inderveer Chana, "Improving Neural Machine Translation Using Rule-Based Machine Translation". 7th International Conference on Smart Computing & Communications (ICSCC),Curtin University, Miri, Malaysia, IEEE, 2019.
- Muskaan Singh, Ravinder Kumar, and Inderveer Chana, "Hybrid Machine Translation System Using Deep Learning". ASM Sc. J., 13, Special Issue 2, 2020 for ICSCC2019, 31-45
- Muskaan Singh, Ravinder Kumar, and Inderveer Chana, "Neural-Based Machine Translation System Outperforming Statistical Phrase-Based Machine Translation for Low-Resource Languages". Twelfth International Conference on Contemporary Computing (IC3), Noida, India, IEEE, 2019.

Chapter 4 *Deployment of Sanskrit-Hindi Machine Translation on cloud platform(CBSHH-MTS)*: This chapter provides the discussion on the deployment of SHH-MTS on the cloud platform. SHH-MTS integrates linguistically-rich approach rule-based with prominent result-oriented approach neural-based and gaining significant attention nowadays. It is a complex application with a large number of modules. Deploying such a complex application on a standalone machine is a time-consuming task. The earlier work struggle with many drawbacks such as slow speed, less data accuracy, and low response time. All these factors adversely affect the performance of the system. A local server takes more time to respond and provides lesser accuracy. Therefore, offering MTS as a cloud service is a better proposition for increasing performance and response time. Auto-tuning for neural-based MT is not possible at the local host due to memory issues, but it is possible on the cloud. Several layers add up automatically to attain maximum accuracy and high speed. The proposed CBSHH-MTS provides better response time, CPU utilization, throughput, rule matching probability and server load. Chapter 4 has been derived from:

- Singh, M., Kumar, R. & Chana, I. A forefront to machine translation technology: deployment on the cloud as a service to enhance QoS parameters.Soft Computing(2020). https://doi.org/10.1007/s00500-020-04923-7

- Muskaan Singh, Ravinder Kumar, and Inderveer Chana. "CDFM-based Secure & Efficient Architecture for Data Management in Cloud Computing." 2019 2nd International Conference on Intelligent Communication and Computational Techniques (ICCT). IEEE, 2019.

Chapter 5 *Validation of the Proposed System*: This chapter demonstrates the verification details, experimental set-up and testing results of the proposed Sanskrit to Hindi hybrid machine translation system. The proposed work has been tested and validated by carrying out a case study of error taxonomy. The performance of the proposed system has been compared with the three standard approaches rule-based,neural-based and hybrid along with their deployment on cloud. The proposed system has been evaluated on automatic measures, i.e., BLEU, METEOR, WER and F-measure and human measures by performing a case study across 15 cases. The results have exhibited that the performance of hybrid is the most satisfactory among the three for processing Sanskrit to Hindi. The performance testing was also carried out on AWS cloud based on the average response time of rule matching probability and number of matching action rules, CPU utilization of packet arrival rate, cost of resources, throughput, time concerning the number of virtual machines and server load concerning time. The hybrid system has performed better on the cloud as well deployed a hybrid machine translation system was deployed on the cloud as a service. Chapter 5 has been derived from:

- Muskaan Singh, Ravinder Kumar, and Inderveer Chana. "Machine Translation Intelligent System as a Service: An aid to Teaching and Learning Process" ,Computer, IEEE Computer Society.[Major Revision]

- Muskaan Singh, Ravinder Kumar, and Inderveer Chana. "A Error Taxonomy for Morphologically-Rich Languages: A Case Study on Sanskrit-Hindi language", Transaction for Asian and Low Resource Language Information Processing, ACM [Under Review]

Chapter 6 *Case Study on the Proposed Machine Translation System*: This chapter outlines the developed taxonomy of error analysis based on different linguistic levels, i.e., orthography, morphology, lexical, syntax, semantic and pragmatics. Consequently, the previous taxonomies were expanded to adapt the error transpired in the morphological rich Indo-European languages. This work substantially provides future research directions and aid in the human error analysis process. The proposed system has examined the errors generated from the rule, neural and hybrid-based approaches. These approaches

exhibited different challenges in performing a case study on the translation of Sanskrit to the Hindi language. In this chapter, an MT system as a service for Sanskrit to the Hindi language has also been developed using linguistic analysis along with deep learning to aid the teaching and learning process. It makes the user free from the cumbersome process of accessing to the dictionary and gaining knowlege of syntax and semantics for Sanskrit language which is a tough and time-consuming process. This interactive interface will assist the school as well as university students enrolled in distance education by promoting self-learning. The main aim of the proposed system is to make the scriptures and philosophical texts such as Gita, Ramayana and Upanishads, available in the Sanskrit language, accessible to the common user.

Chapter 7 *Conclusion and Future Work*: This chapter concludes the work by highlighting its significant contribution to the research. It also provides valuable directions for future directions on the research in the field under investigation.

Chapter 2

Literature Survey

The previous chapter provides a detailed level of view for thesis. It discusses the fundamental concepts related to MT and cloud computing. It further unfolds its close alliance with other underpinning technologies and covers different challenges of this research area. Henceforth, it motivates the research, problem statement, objectives and culminates with the discernment of the contributions and the organization of the rest of the thesis.

The current chapter primarily aims at reviewing the existing literature on the subject under study i.e., Machine Translation(MT). It is a modest endeavour to study in detail the extant modelling techniques of MT. It provides a chance to deeply understand the various issues and aspects of the current study. It also serves the purpose of finding the gap in the research area and avoid duplication. The main objective of this chapter is to assists the developers with resource's required for modelling techniques such as corpus, domains, to toolkits, models, features and their evaluation measures.

This chapter introduces modelling techniques for MT and its details have been organised into following different sections. Section 2.2 contains challenges pertaining to Sanskrit languages. It covers different linguistic and technical challenges. Section 2.3 compares modelling techniques on the resource requirements while Section 2.4 compares MTS for different ancient languages. The synthesis derived from this comparison resulted in minimal work for Sanskrit which is further reviewed in Section 2.5. At the last, this chapter also reviews the MTS on cloud and finally the conclusion of this chapter in Section 2.7.

2.1 Machine Translation Modelling Techniques

This section briefly reviews the pre-eminent MTS modelling techniques. These techniques have been classified based on engineering involved for developing it either manually or mechanically. The techniques have been majorly categorised into human-engineered, machine-engineered, and combination of both, i.e., hybrid. The human-engineered modelling technique is rule-based, while machine-engineered is corpus-based. These techniques have been further classified as shown in Figure 2.1 and reviewed in further subsections.

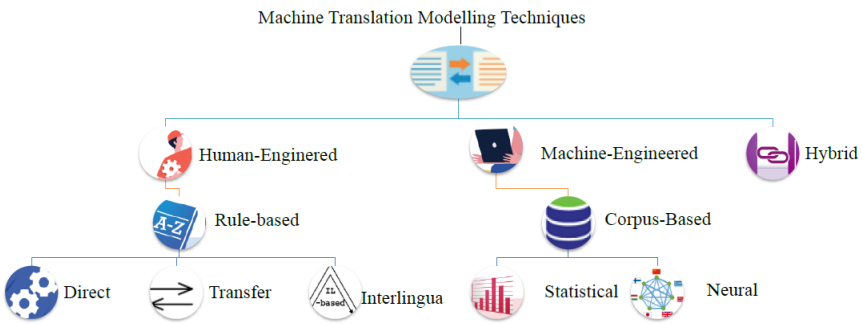


Figure 2.1: Classification of Translation Modelling

2.1.1 Human-engineered Translation

This modelling technique involves rule-based MTS which can be built using dictionary-based, transfer or interlingua approach. It involves more human intervention as all the modules of this system require human insights.

2.1.1.1 Rule-based Machine Translation (RBMT) Modelling

The RBMT modelling technique is one of the oldest and is still being used for less-resourced languages [55]. This technique depends on the linguistic features of the source and target language. The linguistic information along with grammatical properties(morphological, semantic and syntax) is acquired from lexical resources such as bilingual, unilingual or multilingual corpus, dictionaries and rules. As provided with the source sentence, it is processed through many linguistic phases for collecting all the grammatical information, and then words are disambiguated to generate the target sentence. The syntax is mapped with the help of parsing. It depends on colossal lexicons and linguistic rules. The translation quality can be enhanced by adopting contextual reference of a word in the sentence. This technique supersedes the MTS default settings of linguistic rules of each word. The MTS using the rule-based technique can be built using the dictionary, transfer and interlingua approach.

i *Dictionary-based MT*: This technique provides direct transfer of meaning from source to target language words based on dictionary entries. It does not cover syntactic structure and semantic information of the language pair. It is one of the simplest and easiest techniques. It identifies the root word with the removal of suffixes and look-up the bilingual dictionary for the meaning of source word into the target word. The final output can also be post-edited by re-ordering the sentence[11]

ii *Transfer-based MTS*: This technique transfers the source language parse tree into target language parse tree. It covers the syntactic aspect of the language pairs. It performs the analysis of source language structure and transfers it to the target language, and finally, generates the target sentence[5].

iii *Interlingua-based MTS*: This technique converts the source language into an intermediate or interlingua structure to generate target translation. It analyses the source sentence and performs synthesis to convert it into the target sentence. The idea is to capture the meaning in the interlingua. It is independent of the language pair. The transfer of interlingua is performed on the semantic level as well as syntactic level [56].

RBMT is the oldest modelling technique applied to develop MTS. Some of the recent research works using a rule-based approach have been reviewed along with the evaluation measures as shown in Table 2.1. Table 2.2 exhibits methodology used by these RBMT systems along with their domains. The analytical study performed for MTS build using a rule-based approach concludes that most of RBMTs are for English-Hindi language, and English to Marathi as shown in Figure 2.2.

Table 2.1: RBMT Systems based on their Language Pair with the Respective Accuracy

Author	Year	Language	Parameters/Accuracy	
			BLEU	Accuracy
Kavirajan et al.[57]	2017	English - Tamil	NA	71.80
Rana and Antique[58]	2016	English - Hindi	NA	81.70
Darbari et al.[59]	2015	English-Hindi	64.6	NA
G V Garje et al. [60]	2014	English-Marathi	44.29	49.78
Chethan Basavaraddi et al.[61]	2014	English-Kannada	NA	NA
Abhay Adapanawar et al. [62]	2013	English-Bengali	NA	82.92
Abhay Adapanawar et al.[62]	2013	English-Marathi	NA	NA
Devika Pisharoty et al.[63]	2012	English-Marathi	NA	NA
Latha R Nair[64]	2012	Malayalam-English	NA	NA
Batra and Lehal[65]	2010	Punjabi-English	NA	85.33
Rajan et al.[66]	2009	English - Malayalam	NA	79.60
Sinha[67]	2009	English-Hindi	34.12	NA
		English-Urdu	35.44	NA
Sinha and Jain[68]	2003	English-Hindi	NA	90
Shachi Dave et al.[6]	2001	English-Hindi	NA	NA

Table 2.2: RBMT System Methodology Adopted with its Specific Domains and Corpus

Author Name	Year	Methodology	Domain	Corpus
Kavirajan et al.[69]	2017	Sentence simplifier, linguistic rules	NA	250 sentences
Rana and Antique[58]	2016	Fuzzy Rule-based Translation	NA	50,100,200,300,500,1000 sentences
Darbari et al.[59]	2015	TAG & MCSSG Approach	Rajya Sabha using Tree Adjoining Grammar (TAG)	Rajya Sabha website since 2006
Chethan Basavaraddi et al.[61]	2014	Re-formatting, pre-editing, morphological analysis, transfer of internal representation and generation along with reformatting	Goverment and education sector	NA
G V Garje et al. [60]	2014	Pre-translation processor,parsing, Named Entity Tagger, rearrangement generator, sentence filter,word by word translation,disambiguator, target generator	Tourism	1000 sentences from TDIL
Chandranath Adak[70]	2014	Morphological analysis, Lexicalization:POS tagging, Re-ordering, Transliteration and combine	General	2574 words
Dubey[71]	2014	Direct technique	NA	18500 words
Devika Pisharoty[63]	2012	Tokenization, lemmatization, parsing, syntax validation, semantic validation, translation, transformation and reconstruction of sentence	NA	NA
Latha R Nair[64]	2012	Pre-processing, morphological parser, transfer and generator	NA	NA
Abhay Adapanawar et al.[62]	2013	Tokenization, POS tagging, dictionary look-up and rule extraction from database	Assertive sentences	NA
Batra and Lehal [65]	2010	Three Components-based approach: analyzer, a transfer component, and a generation component	NA	500 sentences
Rajan et al.[66]	2009	Transfer link rules, Morphological Rules	Word Dictionary	5000
Sinha[67]	2009	Interlingua based rule-based approach.	NA	100000 Words
Sinha and Jain[68]	2003	Rule-based	NA	NA
Shachi Dave et al.[6]	2001	English analyzer with disambiguation, UNL conversion and Hindi generator	Political and stock market stories	180 sentences(Ministry of Information and Technology) and Brown corpus.

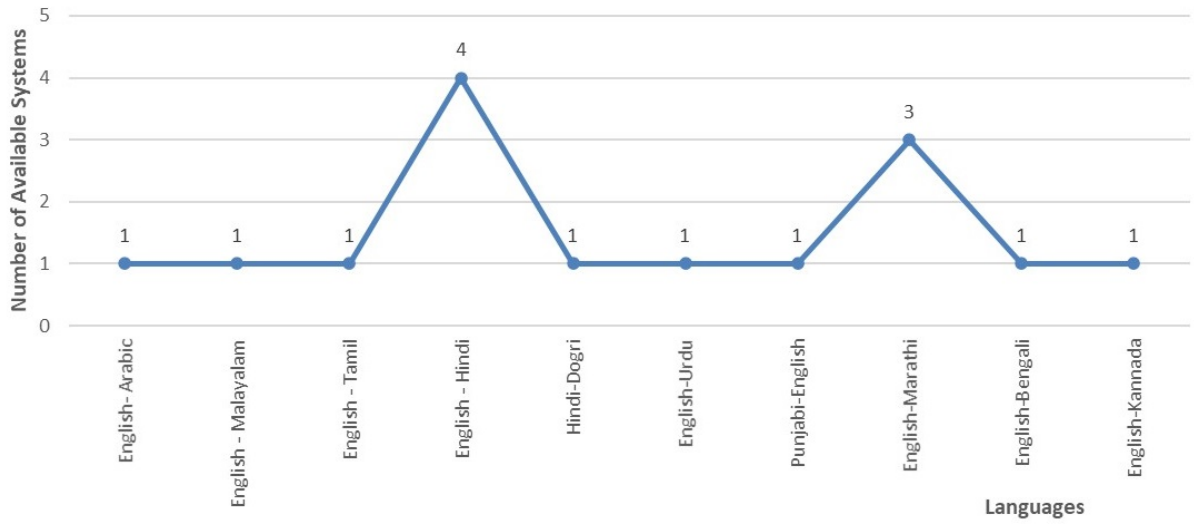


Figure 2.2: Rule-based Modelling for Different Languages

2.1.2 Machine-engineered Translation

These modelling techniques involve more machine intervention as compared to human-engineered as all the modules of the system require machine processing. It builds a translation model using machine learning from the corpus (monolingual and bilingual). The machine-engineered MTS includes a corpus-based modelling technique which has been further classified into statistical and neural-based.

2.1.2.1 Corpus-based MTS

The approach depends entirely on the corpus (bilingual, multi-text or parallel corpus). MTS can be developed by training the corpus to perform translation. In comparison to the rule-based modelling technique, it requires fewer efforts as it is more machine-dependent. This technique has been further classified into statistical and neural-based.

I Statistical Machine Translation (SMT) Modelling Technique

It produces translation-based on the statistical model while learning from parallel and monolingual corpus. It was introduced in the late 1950s[72] and had been performing quite well until the present time. The statistical model assumes the presence of aligned large quantity and high-quality data. It encodes the data information in the language and translation model which is decoded by the decoder to generate the translation. The models of language and translation are produced using the given source and target sentence. The language model is built using the monolingual corpus of the target language. It also assigns the probability to each string by calculating the relative frequencies. While the translation model is built using parallel corpora by assigning probability, i.e., the source sentence is a translation of the target sentence. Mathematically, using Bayes theorem in Eq.(2.1)[7],

$$P(s|t) = \frac{P(s) \times P(t|s)}{P(t)} \tag{2.1}$$

The highest probability sentence is chosen as the best translation using Eq.(2.2)

$$e_{max} = \hat{argmax} P(s)P(t|s) \tag{2.2}$$

In the decoding phase, a source sentence chooses a translation sentence with maximum probability as in Eq.(2.3),

$$argmax P(t|s) = argmax P(t) \times P(s|t) \tag{2.3}$$

The decoding phase substitutes the phrases from left to right to produce the translation. It provides a dynamic programming solution by applying the beam search algorithm. The fluency of translation output depends on the language model, and adequacy of translation is derived by translation model. Though decoding is a complex process, it provides the output sentence along with re-ordering .
There are different models of SMT based on segmentation of source sentence into words[73][74], phrase[75], syntax[76][77] and hierarchy[78]. The SMT systems utilise data and human resources in a better way. It is better than rule-based as it does not require human-formed linguistic rules which are time-consuming and language-specific. It demonstrates to have higher productivity and quality for domain-specific translations such as news, official documents and literature.

The survey performed on some of the recent research works using the statistical approach has been presented in the subsequent tables. Table 2.3 compares the various SMT-based MTS in terms of different evaluation measures used for computation of accuracy. SMT systems are developed using different methodology having several toolkits. These are described in Table 2.4. The corpora details along with their domains are mentioned in Table 2.5. The analytical study performed for Indic languages using SMT modelling technique has been shown in Figure 2.3 for various languages, and concluded that maximum work has been done to translate different languages into English and Hindi.

II Neural Machine Translation(NMT) Modelling Technique
NMT is the most recent technique for MT and is said to make a substantially more accurate translation. It depends on the model of neural systems in the human cerebrum and with data being sent to various "layers" to handle before output. It utilises deep-learning procedures to guide itself to translate content given existing

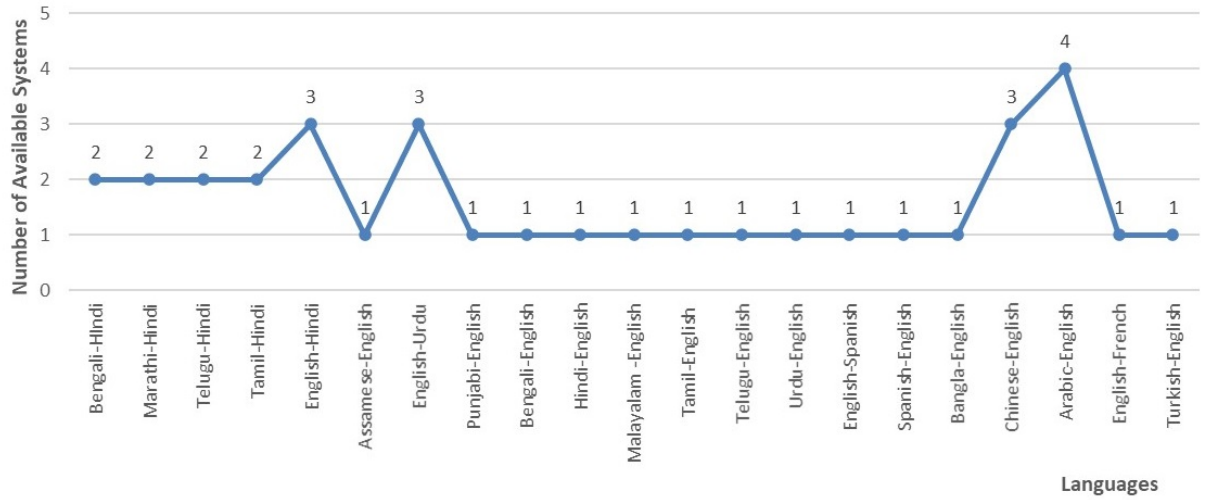


Figure 2.3: Statistical Modelling for Different Languages

reference translation and build models. It forms the technique faster as it requires a single sequence model rather than multiple models as in SMT. It also produces higher quality output. It models the source sentence $s_1, s_2, s3.....s(m)$ to target sentence $t_1, t_2, t3.....tn$ with conditional probability modelling context vector c as in Eq(2.4)

$$logP(t|s) = \sum_{i=1}^{n} logP(y_i|y < i, c) \tag{2.4}$$

The basic form of NMT consists of encoder and decoder components. The encoder encodes the source sentence into a context vector c, while decoder decodes this context vector and generates one word at a time. NMT requires minimal domain knowledge. The sequence to sequence learning proposed by Sutskever et al., [103] and Cho et al., [15] employed by Luong et al., and Jean et al., [104][105], i.e., reads a sentence till the end and output one word at a time. It produces good translation results. It has the drawback of encoding source sentence into a fixed-size vector which deteriorates in quality when exposed to longer sentences. However, this drawback can be overcome by attention mechanism[8]. There are two different architectures for constructing NMT.

i *Recurrent Neural Network(RNN)*: It has been producing good quality translation results. RNN is composed of encoder and decoder with a similar working of sequence to sequence learning. Different RNN architectures are experimenting different models, [14][106][107][108][105][109] [110].

ii *Convolution Neural Network(CNN)*: It has achieved surpassing results for the word-based MTS, but along with RNN [14][111]. These works have applied convolution layer on the bottom of the recurrent layer which hinders the performance. The bottleneck was handled by implementing the fully convolutional

Table 2.3: SMT System for Languages with their Respective Accuracy

Author	Year	Language	Parameters/Accuracy		
			BLEU	NIST	TER
Subalalitha et al.[79]	2018	English-Hindi	NA	NA	73.43
Shishpal Jindal[80]	2018	English-Punjabi	87.67	NA	NA
Raj Nath Patel et al.[81]	2018	English-Malayalam	8.25	NA	21.57
		English-Hindi	19.43	NA	37.77
		English-Punjabi	23.09	NA	44.06
		English-Tamil	7.56	NA	23.62
Khan et al.[82]	2017	Bengali-English	19.70	3.78	NA
		Hindi-English	19.30	37.79	NA
		Malayalam -English	11.10	NA	NA
		Tamil-English	12.80	NA	NA
		Telugu-English	14.20	NA	NA
		Urdu-English	24.70	4.26	NA
Patel et al.[83]	2016	Bengali-Hindi	31.3	6.80	46.06
		Marathi-Hindi	38.71	7.36	41.38
		Telugu-Hindi	28.51	6.26	50.80
		Tamil-Hindi	19.19	5.01	62.90
		English-Hindi	20.75	5.61	61.70
Patel and Pimpale[84]	2016	Bengali-Hindi	33.77	7.19	45.52
		Marathi-Hindi	41.20	7.93	39.25
		Telugu-Hindi	29.72	6.66	49.26
		Tamil-Hindi	20.38	5.34	62.25
		English-Hindi	24.00	6.12	58.99
Das and Baruah[85]	2014	Assamese-English	11.32	NA	NA
Ali et al.[86]	2014	English-Urdu	9.03	NA	NA
Khan et al.[87]	2013	English-Urdu	2.90	6.36	NA
Ali et al.[88]	2013	English-Urdu	37.10	NA	NA
Kumar and Kumar[89]	2013	Punjabi-English	NA	97.00	NA
Aasim Ali et al.[86]	2010	English-Urdu	9.03	NA	NA
Anwar et al.[90]	2009	Bangla-English	NA	92.53	NA
Udupa and Faruquie[91]	2005	English-Hindi	13.44	4.57	NA

model as suggested by Kaiser et al., [112] and Kalchbrenner et al., [113]. It later formed stacked convolutional layers[114] with GPU version of the neural model. The performance and accuracy was improved with a number of models [113],[115][116][114].

NMT makes it easier to train large models and generalise long sentences. It doesn't have to store phrase tables, language models, score tables as in SMT.

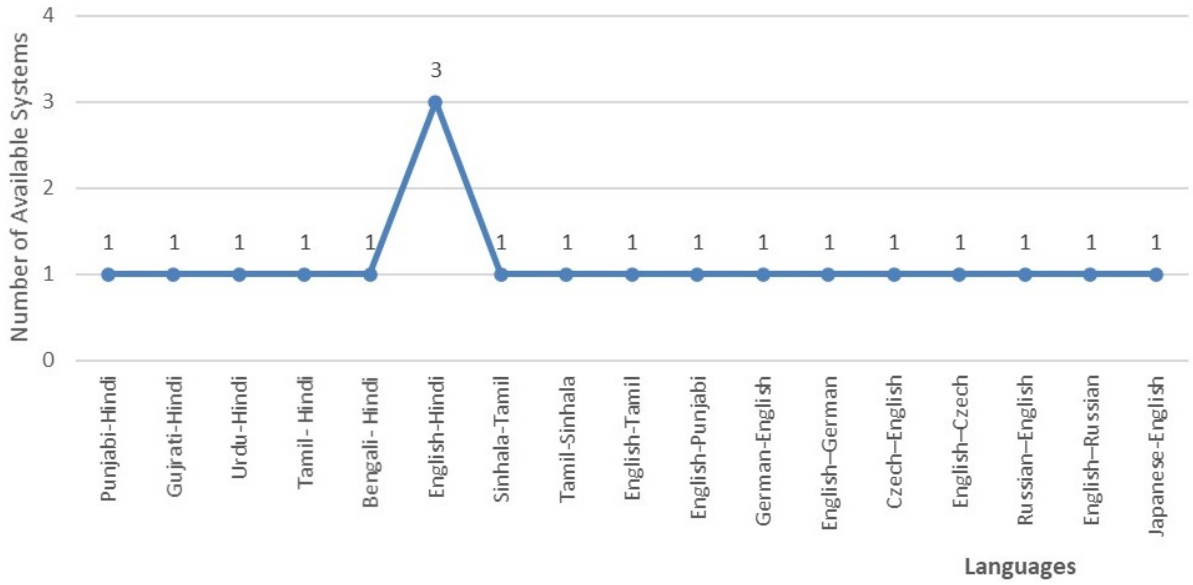


Figure 2.4: Neural Modelling for Different Languages

The survey performed on some of the recent research works using a neural-based approach has been presented in the form of subsequent tables. There are different toolkits available for developing the NMT system experimenting with methodologies as shown in Table 2.6. The experiment is conducted on different corpora and domains mentioned in Table 2.9. These systems are evaluated with different measures such as automatic or human, which have been presented in the Table 2.7. The analytical study performed for Indic languages using neural-based techniques has been shown in Figure 2.4, and concludes that English to Hindi translation system is more prominent as compared to other languages. It is because more parallel corpora exist for this language pair than other Indic languages.

2.1.3 Hybrid Machine Translation(HBMT) Modelling

HBMT is a preferred translation technique as it combines the best of human-engineered and machine-engineered approach. It is characterised by the use of multiple MT modelling techniques within a single MT system. The motivation for developing a hybrid approach stems from the failure of any single technique to achieve a satisfactory level of accuracy. Many HBMT systems have been successful in improving the accuracy of the translation systems.

The HBMT architecture is guided by the human-engineered approach, i.e. rule-based using corpora to build[[117][118][119][120][121][122][123][124]], using corpus-based tool for weighing the RBMT output[[125] [126][127][128][129][130]], RBMT is guided by statistical post-editing[[17][131][132][133]]. Corpus-based HBMT uses rules at pre-processing and post-processing suggested by

[[134][135][136][137][138][139]], incorporating dictionaries and rules in corpus-based MTS [[140][141][142][143][144][145][146][147]] and building HBMT using corpus with statistical approach[[148][18][149][150] [151] [152] [153]]. These several works demonstrate that HBMT provides better translation quality. Most of the techniques concatenate rules and data, whereas fewer works are combining machine-engineered approaches. Some of the recent work incorporates additional information to guided RBMT or guided corpus-based technique. The hybridisation technique has grown to speech translation, cross-lingual information retrieval and computer-aided and post edited systems. The survey performed on some of the recent research work using the

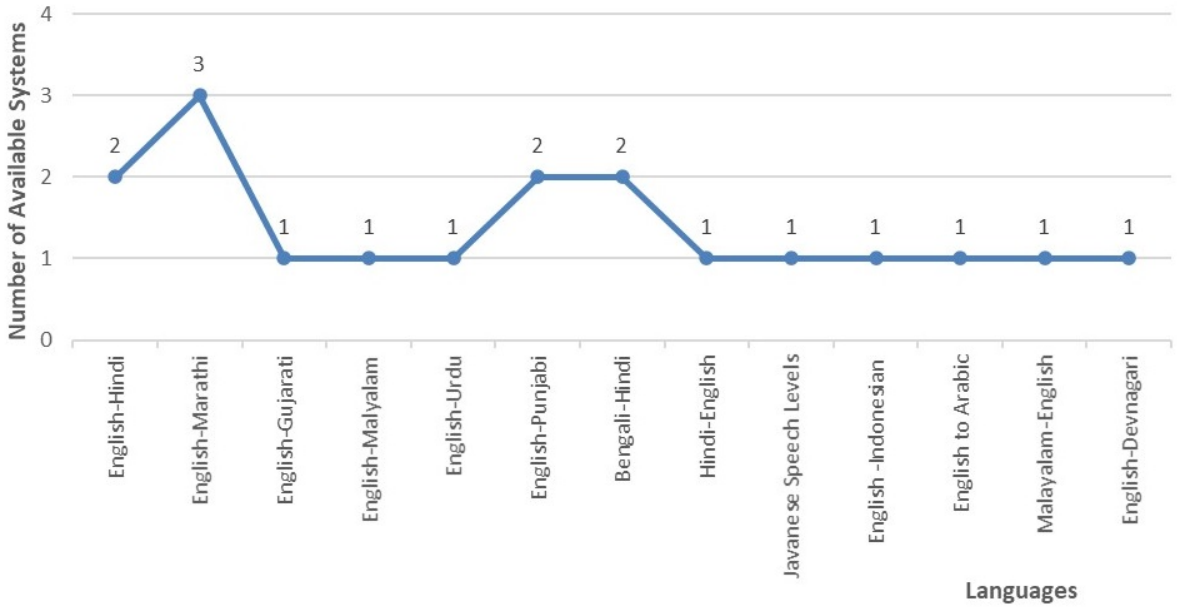


Figure 2.5: Hybrid Modelling for Different Languages

hybrid approach has been presented in the form of subsequent tables. The HBMT developed for Indic languages using different experiment or methodology along with toolkits is depicted in Table:2.10. These system developed with corpus along with domains is mentioned in the Table:2.11. Evaluating these systems with different evaluation measures is shown in Table:2.8. The analytical study performed for Indic languages using hybrid approach has been shown in Figure.2.5 and concludes that English to Marathi, English to Hindi, Bengali to Hindi, and English to Punjabi have notable more systems as compared to other languages.

Table 2.4: SMT Models along with Methodology and Toolkit

SMT Model	Author	Year	Methodology	Toolkit
Phrase-based	Subalalitha[79]	2018	n-gram and Naive Bayes Probability	NA
	Shishpal Jindal[80]	2018	IRSLTM language model,Trained model with GIZA++ alignment and testing	Moses,IRSLTM and GIZA++
	Raj Nath Patel et al.[81]	2018	Pre-processing, re-ordering, suffix separation, transliteration	Moses,MERT, KenLM and Kneser-Key.
	Khan et al.[82]	2017	Sampling,Tokenization,Tuning Set	Moses, GIZA++, Kneser-Ney, SRILM
	Patel and Pimpale[83]	2016	Pre-Processing, Transliteration	Modified KenLM, Moses.
	Patel et al.[84]	2016	Suffix Separation (Compound Splitting and Reordering)	Modified KenLM
	Ali et al.[86]	2014	Manual Alignment,Translation and Tuning	Moses, GIZA++
	Das and Baruah[85]	2014	Text, Decoder and Transliteration	IRSTLM tool, GIZA++, Moses Decoder
	Ali et al.[88]	2013	Tokenization, Training and Tuning	Moses, IRSTLM
	Kumar and Kumar[89]	2013	N-Gram model and Transliteration	NA
	Aasim Ali[86]	2010	Data distribution, mkcls, GIZA++ and MERT tuning	Moses, GIZA++ and MERT
	Anwar et al.[90]	2009	Tokenization, Syntax Analysis, Parsing and NLP Conversion	NA
	Anwar et al.[90]	2009	Tokenization, Syntax Analysis,Parsing and NLP Conversion	NA
	Cherry[92]	2008	NA	Moses, GIZA++,and Phrasal
	Udupa and Faruquie[91]	2005	NA	Moses, GIZA++,and Phrasal
	Udupa and Faruquie[91]	2005	NA	IBM models 1, 2, and 3.
Hierarchical-based	Khan et al.[82]	2013	Sampling, Tokenization and Tuning	Moses Decoder, GIZA++, SRILM

Table 2.5: SMT Classification-based on Different Domains and Corpuses

Author Name	Year	Domain	Corpus
Subalalitha[79]	2018	News, Agriculture and Technical phrases	IIT Bombay and manually collected corpus.
Shishpal Jindal et al.[80]	2018	Health,Tourism and Gyan Nidhi	Manually curated.
Raj Nath Patel[81]	2018	General	MTIL-2017
Khan et al.[82]	2017	Consumer, education, health, housing, legal and social documents	EMILLE
Patel et al.[83]	2016	Health, Tourism, and General Domain	ILSMT and 23K sentences from other
Patel and Pimpale[84]	2016	Health, Tourism, and General Domain	ILSMT, 23K, and 500 sentences from other
Das and Baruah[85]	2014	Tourism Data	parallel corpora of about 8000 sentences
Ali et al.[86]	2014	Quran, Ahadeeth	6000 sentences
Aasim Ali[86]	2010	Religious	Adaheeth
Khan et al.[82]	2013	NA	EMILLE
Ali et al.[88]	2013	NA	41208 sentences
Kumar and Kumar[89]	2013	Names	15000
Anwar et al.[90]	2009	NA	NA
Udupa and Faruquie[91]	2005	News, government documents, conversation, and magazine articles.	150,000 sentence pairs

Table 2.6: NMT System based on Toolkit with its Respective Methodology

Toolkit	Author Name	Year	Methodology	Model
TensorFlow	Saurav Jha et al.[93]	2018	Sequence to Sequence, Alignment, hierarchical Attention Network, Transformer Network and Character Encoding	Bi-LSTM with Attention.
GoogleTranslate API	Himanshu Choudhary et al.[94]	2018	Sequence to Sequence Model, Attention, BPE, Word Embedding	Bi-LSTM with Attention
OpenNMT	Pathak and Pakray[95]	2018	Data Pre-processing, Training, Encoding, Decoding and Translation	NA
OpenNMT	Ramesh and Sankaranayanan[96]	2018	NA	Attention-based model
TensorFlow	Singh et al.[97]	2018	Neural Machine Translation,Training, Testing	LSTM with attention.
Nematus	Jigar Mistry[98]	2017	Encoder-decoder model, Attention and BPE	Vanilla LSTM or GRU
OpenNMT	Revanuru et al.[99]	2017	Input, Bi-LSTM, Sum, LSTM, Bridge and Decoder	Deep Bi-LSTM
OpenNMT	Tennage et al.[100]	2017	Pre-processing, OpenNMT, Benchmark, Training and Word Phrases	NA
TensorFlow	Aggarwal and Sharma[101]	2017	LSTM and GRU Cells, Encoders and Decoders	Bi-LSTM with attention mode
Theano	Yerra et al.[102]	2016	NA	Attention-based model

Table 2.7: NMT Systems with their Language Pairs and Respective Accuracy

Author Name	Year	Language	Parameters/Accuracy	
			BLEU	Accuracy
Saurav Jha et al.[93]	2018	Hindi-Bhojpuri	90.89	90.23
Himanshu Choudhary[94]	2018	English-Tamil	8.33	NA
Pathak and Pakray[95]	2018	English-Hindi	52.54	NA
Ramesh and Sankaranayanan[96]	2018	English-Tamil	5.53	NA
		English-Hindi	3.97	NA
Singh et al.[97]	2018	English-Punjabi	26.07	NA
Jigar Mistry[98]	2018	English-Hindi	26.88	NA
		Bengali-Hindi	33.87	NA
		Gujarati-Hindi	53.95	NA
Revanuru et al.[99]	2017	Punjabi-Hindi	46.47	NA
		Gujarati-Hindi	35.69	NA
		Urdu-Hindi	22.47	NA
		Tamil- Hindi	7.56	NA
Aggarwal and Sharma[101]	2017	English-Hindi	9.23	NA
Tennage et al.[100]	2017	Sinhala-Tamil	7.50	NA
		Tamil-Sinhala	12.75	NA
Yerra et al.[102]	2016	Bengali- Hindi	20.41	NA

Table 2.8: Hybrid MT based on Language with its Respective Accuracy

Author	Year	Language	Parameters/Accuracy	
			BLEU	Accuracy
Dhariya et al.[154]	2017	Hindi-English	NA	86.50
Salunkhe et al.[155]	2016	English -Marathi	NA	83.00
B and Joseph[156]	2013	English-Malayalam	69.33	75.30
Kaur and Laxmi[157]	2013	English-Punjabi	NA	81.67
Dhore and Dixit[158]	2011	English-Hindi	NA	97.25
		English-Marathi	NA	97.00
		English-Gujarati	NA	96.50
Chatterji et al.[159]	2011	Bengali-Hindi	29.45	NA
Shahnawaz and Mishra[160]	2011	English-Urdu	69.54	NA
Chatterji et al[161]	2009	Bengali-Hindi	22.57	NA

Table 2.9: NMT Systems with their Respective Domain and Corpus

Author	Year	Domain	Corpus
Himanshu Choudhary et al.[94]	2018	News, Bible, Cinema, Movie Subtitles	EnTamV2.0 and Opus
Saurav Jha et al.[93]	2018	Dictionary	Manually curated words from dic.
Pathak and Pakray[95]	2018	General Domain	MTIL
Ramesh and Sankaranayanan[96]	2018	Wikipedia	Wikimedia dumps
Singh et al.[97]	2018	Health, tourism, agriculture and entertainment	TDIL,EMILLE,OPUS
Revanuru et al.[99]	2017	Agriculture, entertainment, health and tourism	TDIL-DC
Aggarwal and Sharma[101]	2017	General Domain	50,000 sentences from Bojar corpus and ILCI
Tennage et al.[100]	2017	Annual reports, establishment codes, order papers, and official letters	official government documents of Sri Lanka
Mistry Jigar et al.[100]	2017	Health and tourism	ILCI

Table 2.10: Hybrid MTS Modelling Technique based on its Model and Toolkit

Technique	**Author**	**Year**	**Model**	**Toolkit**
Rule-based+Statistical-based	Salunkhe et al.[155]	2016	NA	Open NLP
Rule based +Statistical Phrase based	Dhore and Dixit[158]	2011	NA	NA
Rule-based (lattice-based lexical transfer) +Statistical-based	Chatterji et al.[159]	2011	Phrase-based	NA
Rule-based (Lexical transfer-based) +Statistical-based	Chatterji et al.[161]	2009	Phrase-based	GIZA++
Statistical Machine Translation + Translation Memory.	B and Joseph[156]	2013	Phrase-based	IRSTLM, GIZA++, Moses decoder
Rule-based+Example-based	Kaur and Laxmi[157]	2013	NA	NA
Rule-based +Neural-based	Shahnawaz and Mishra[160]	2011	NA	Java(jdk1.5) with Matlab 7.1
Rule-based + Example-based+ Statistical-based	Dhariya et al.[154]	2017	Phrase-based	NA

Table 2.11: Hybrid Systems with their Domain and Corpus

Author	**Year**	**Domain**	**Corpus**
Dhariya et al.[154]	2017	NA	CFILT, IIT-Bombay
Salunkhe et al.[155]	2016	NA	Parallel Corpus(IIT-Bombay)
B and Joseph[156]	2013	Indian history and Islamic history	563 sentences
Kaur and Laxmi[157]	2013	News headlines	300 sentences
Dhore and Dixit[158]	2011	Banking glossary	Reserve Bank of India
Chatterji et al.[159]	2011	Tourism	2000 sentences
Shahnawaz and Mishra[160]	2011	NA	NA
Chatterji et al.[161]	2009	Written	EMMILE-CIIL

2.2 Challenges of MTS for Processing Sanskrit Language

This section reviews the types of challenges that one must consider in developing an MTS. These challenges have been examined along with two dimensions; the first on different types of linguistic considerations (e.g. syntactic word order and semantic ambiguity), and the second on different types of operational or technical considerations.

2.2.1 Linguistic Challenges

The pattern of divergence between two languages needs to be recognised before building an MTS. The challenges for translation are specific to language pair as it is difficult to build a general approach applicable to all languages. The work studying the deviation of languages by Dorr [162][163] has formed a basis for further research for Indian languages. The challenges to translate English-Sanskrit-Hindi MT[164], English-Hindi[6] and English-Sanskrit[165] have also been considered. These linguistic challenges are specific to Sanskrit language and need to consider before building MTS for Sanskrit to English language or Sanskrit to the Hindi language.

- Sanskrit contains complex or compound words being influenced by oral tradition, i.e., continuous strings of characters without word boundaries or punctuations. It becomes difficult to guess the boundaries as they undergo euphonic changes[166].
- In Sanskrit, there is a special category of verbs requiring special treatment termed as thematic divergence. The subject Noun Phrase(NP) in Sanskrit and Hindi is the dative case, while in English is the nominative case which causes a divergence in translation[164].
- Sanskrit is rich in inflectional and morphology. This richness makes a difference in the last character of the word, its gender and makes it difficult to remember different forms of word inflections[166].
- The vocabulary of almost all the Indian languages has been derived from Sanskrit. There have been cases of meaning shift, reduction and expansion. It makes it difficult to understand the Sanskrit text without the prior knowledge of original meaning. There are various trends in Sanskrit literature for commentaries. The presentation of commentaries in the Sanskrit language is in a nested form which makes it difficult to understand for modern scholars[166].
- The structure difference between languages leads to problems in translation. Sanskrit and Hindi are Vibhakti and Karaka-based languages while in English whenever Noun Phrase is encountered Vibhakti place is replaced with prepositions or null[164].

- Sanskrit consists of complex words resulting in the translation of two to three words of English in the one-word translation of Sanskrit. This Conflational and Inflation divergence is encountered for Sanskrit and Hindi translation as well[164].
- Though Sanskrit is a free-word order language, still, there are cases where the change in the word order changes the meaning of a sentence for translation of language pair Sanskrit and English[164].
- Sanskrit and Hindi language pairs have most commonly passive voice while English has active voice sentences. This change of voice leads to problems in translation for specific language pair[164].
- There is a categorical and lexical divergence for Sanskrit-English and Sanskrit-Hindi language pairs. The categorical divergence occurs in case of mismatch in parts of speech of translation pair languages. In the translation process when an exact match is not mapped from one language to another leads to lexical divergence. In Sanskrit, a different meaning are generated with the addition of upsarga to the verb[164].
- In the translation of Sanskrit to English/Hindi, there are different adjuncts and clauses which cause divergence in translation. This divergence changes the sentence construction of language pair[26].
- Sanskrit has honorific features containing verb endings with adjectives and nouns. These plural verb and plural pronoun infections are caused because of socio-cultural aspects of languages[26].
- The mapping of time from English to Sanskrit and Hindi causes a problem as a.m. and p.m. in English cannot be mapped in Sanskrit as afternoon and morning. The terms a.m. and p.m. cannot be translated as such in Sanskrit and Hindi[26].
- There is formal grammar defined by Panini for Sanskrit but no such parallel grammar exists for Hindi or other European languages. In the absence of such parallel grammar, exceptional cases are covered by forming linguistic rules. There are cases where Vibhakti diverges from Sanskrit to Hindi such as optional, exceptional, differential, alternative, non-Karaka, verbal and complex-predicate divergence[26].

2.2.2 Technical Challenges

This section aims at studying the technical challenges of Sanskrit-based MTS. The challenges are encountered while developing and applying modelling techniques for MTS.

- Domain Mismatch: In NMT-based system, if the content is not from the same domain, then it exhibits poor performance. It has a low quality for out of domain text; as for

fluency, it sacrifices adequacy. The translation misguides the user by visualising the fluent output in case of information gisting[167][168].

- Amount of Training Data: It performs well for high-resource languages as compared to low-resource languages as its learning depends on the amount of training data. NMT system trains millions of data, showing direct proportionality to accuracy [169][170].

- Noisy Data: NMT system is not robust to noisy data in corpora such as misaligned sentences[171], poorly translated sentences, and content in wrong languages. In such cases, NMT fails to predict the relationship between the language model and the input data context. Even varying the training ratios, the problem does not dissolve as it produces inadequate output.

- Word Alignment: Aligning input words to output words was served by the attention model of NMT[8]. The performance of attention-based NMTs is very poor in case of more substantial sentences, and it does not provide accurate word alignment. Incorporating discrete translation and lexicon dictionaries[172] for improving system with fertility and coverage modelling[173]. The attention mechanism may produce better alignment if provided with guided learning[174][175]. The source sentence can be translated into many target sentences in SMT as well. Aligning of phrases in the target concerning source becomes cumbersome. An efficient alignment algorithm is required after translation. There are several techniques such as a template for aligment[148][176], Hidden Markov Model(HMM)[177], toolkits [178][179][180]. Despite various attempts to enhance alignment in SMT systems, it does not fulfil the role and shows divergence.

- Beam Search: The decoding process generates the translation of highest probability. For highest probability translation, a search operation performed from all possible translations. In NMT, the size of all possible translation is termed as beam size for each input word. It is not directly proportional to the accuracy of the system as after a point it starts degrading. It requires manually normalizing of scores by the sentence length. Falling out of the optimal range(30-50) of beam size the translation quality starts degrading. The wider beams deteriorate the quality with shorter translations[181].

- Longer Sentences and Inflectional Category Words: In complex and large sentences, quality is low as compared to small sentences. Even words which are of low frequency are not easily translated with NMT-based Systems[181].

- Parallel Corpus: Corpus-based modelling techniques require a large amount of parallel and monolingual corpus but it is costly and time-consuming. Training data is directly proportional to the model constructed. A collection of millions of monolingual sentences yields a better language model producing fluent output. The translation model is trained using parallel corpus producing an adequate translation. Even with huge

corpora, translation quality is very coarse. This modelling technique does not apply to many Indic languages as there is very less or no parallel corpora for some low-resource languages[181].

- Idioms and Multiword Expressions: The properties exhibited by idioms and multiword expressions make it difficult for them to translate. The corpora should be customised for a specific idiom for a particular language pair to be most effective. Even then, SMT requires pre-processing and post-processing steps for handling such cases[13].

- Time Consuming and Expensive: SMT and RBMT can also be expensive as they requires a lot of upfront costs. In this, both the processes, i.e., pre-processing and corpus creation are expensive and time-consuming. It also requires collaboration with computer scientists, translators, linguists and statisticians[182].

- Learning: For the learning of the system, every phase requires continuous error detection. It is harder to fix mistakes in the system once they have been implemented. With models like RBMT, you can fix errors and remove certain words quite easily. With SMT, you need to retrain the whole system and check if other errors have emerged or not[13].

- Linguistic Issues: Even training the SMT system with 100 million words, it produces a partially excellent translation. It suffers from various linguistic issues such as mangled grammar, wrong word choices, name translation, unknown words and syntactic transformations[183]

- Linguistic Knowledge: Some linguistic information still needs to be set manually (such as rules, part of speech). This requires human intervention and linguistic knowledge of the source and target language[183].

- Generalized System: It is hard to deal with rule interactions in big systems, resolve ambiguity and handle idiomatic expressions. It is difficult to build a generalized system handling all the aspects of a language pair[182].

- Domain Adaptability:RBMT systems provide a mechanism to adapt to new data with the help of rules and lexicon but it is time-consuming effort. Adapting to new domains also requires extensive knowledge of language and human effort as each word requires a rule to disambiguate its meaning[182].

- Resources: The resources required for developing RBMT are linguistic rules, dictionaries, language-specific tools, morph-analyser, parser and generator. It is an expensive process as it requires much human effort and knowledge. There is also a requirement of efficient corpora for a particular language. Some languages do not have sufficient data and preparation of these data is a challenging task and time-consuming process[182].

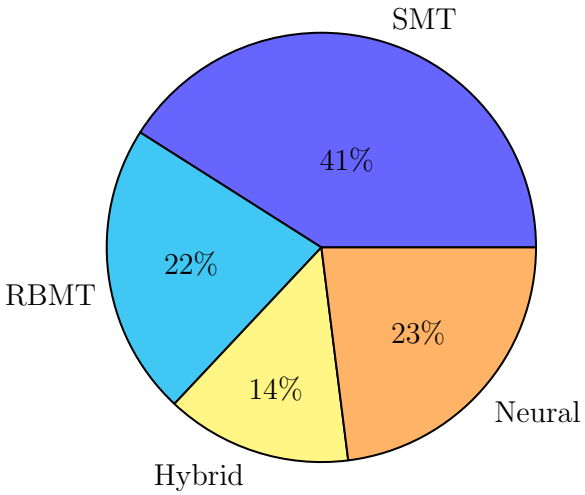


Figure 2.6: Overall Percentage of Various Modelling Techniques

- Modification: Existing RBMT system modification requires to process as an alteration in the rules. It is more expensive as compared to generating new rules[182].

2.3 Comparison of Various Modelling Techniques

In MT research for Indic languages, RBMT and SMT are the methods which are used most frequently. Though RBMT is the oldest approach to achieve good results, the development is very time-consuming as linguistic rules need to be fixed manually for every word in a sentence. In terms of investment, the customisation cycle needs to reach the quality threshold which can be quite long and costly in terms of human resources. The RBMT systems are built with fewer data as compared to SMT systems, along with dictionaries and language rules to translate. It does not produce fluently translated output. Also, language is constantly changing, which means rules must be managed and updated wherever necessary in RBMT systems. Moreover, SMT systems require much less time and linguistic knowledge. SMT models require more computer processing power and storage capacity to build and manage large translation models. Figure 2.6 shows that SMT systems are mostly used by the researchers for the translation of different languages. On the other hand, neural and hybrid-based systems are very less used for Indic language pair due to insufficient parallel data. The performance of these modelling techniques is good with fluent output. The computational requirement for these techniques is more than human resource requirements. Machine learning and linguistic knowledge apply to these techniques to improve performance. These techniques sometimes exhibit out of domain quality. The error analysis is difficult to perform for these techniques. So, these systems need to be tested for future translations. Table 2.12 compares the performance of some of the MT modelling techniques for different quality parameters. The year-wise usage of these techniques is exhibited in Figure 2.7.

Table 2.12: Comparison of MTS Techniques based on Various Parameters

FEATURES	RBMTS	SMTS	EBMTS	NMTS
Performance	Good	Medium	Good	Good
Fluency	Less	Medium	High	Medium
Robust	Yes	No	Yes	No
Human resources requirement	High	Less	Medium	Less
Computational resources requirement	Less	Medium	Medium	High
Machine Learning Applicability	No	Yes	Yes	Yes
Linguistic knowledge requirement	Yes	No	Yes	Yes
Use of grammar	Yes	No	No	No
Out of domain quality	Medium	Low	High	Low
Predict Quality	Good	Similar	Very well	Similar
Consistency	High	Low	Medium	Low
GPU requirement	No	No	No	Yes
Language dependency	Yes	No	No	No
Maintenance	Difficult	Easy	Easy	Easy
Model Size	Huge	Huge	Moderate	Small
Error analysis	Easy	Difficult	Difficult	Impossible
Parallel corpus requirement	No	Yes	Yes	Yes
Dictionary and Rule requirement	Yes	No	No	No
Extendable	Difficult	Easy	Easy	Easy

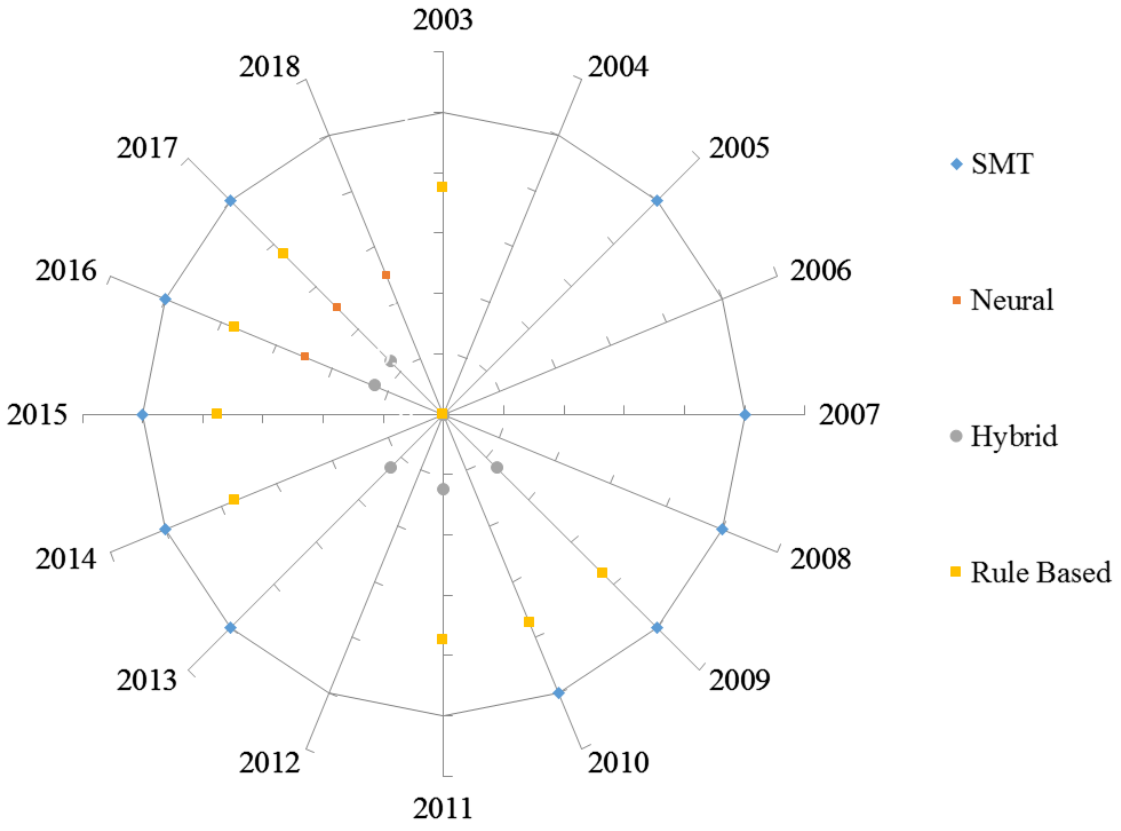


Figure 2.7: Modelling Techniques based on the Year of their Development

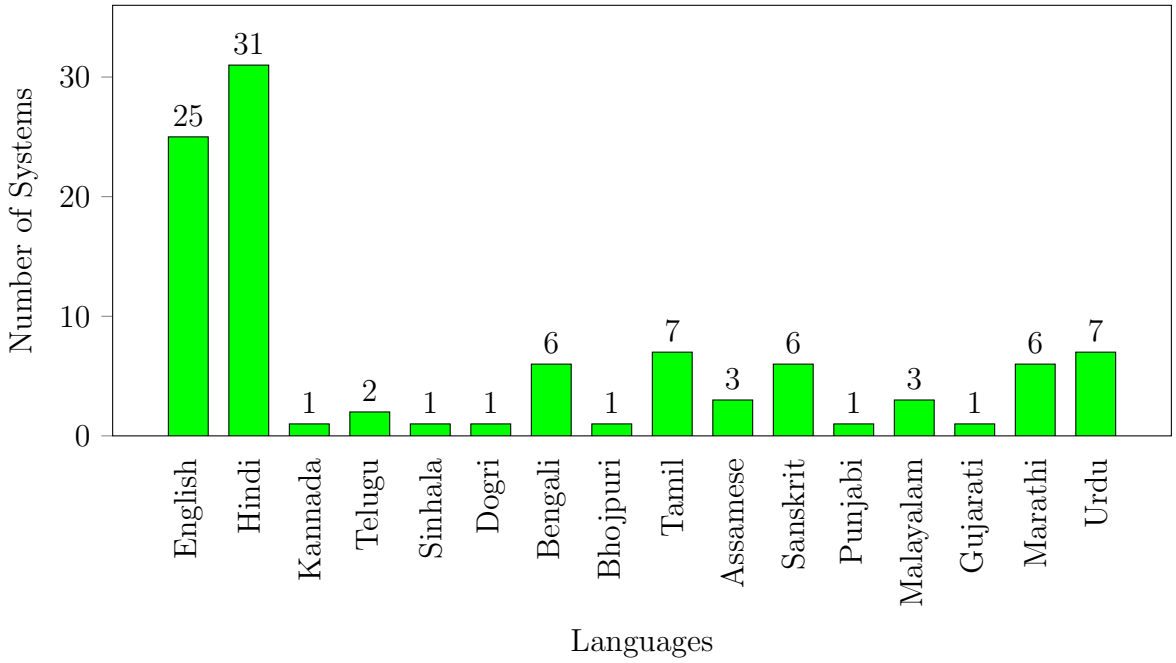


Figure 2.8: MTS Developed for Languages

2.4 Comparison of Machine Translation Systems

MT aims to translate one language to another by utilising different resources. The principal objective is to fulfil the language gap between two distinct languages involving individuals, groups or nations. In India,there are different languages and web-contents extensions which needs massive language translations. The Table 2.13 and Table 2.14 provide the classification of MTS based on human and machine-engineered approach along with their features and outcomes. These tables carry the information like the modelling techniques used, features of the technology, and outcomes along with different language pairs. An analysis of the percentage of work performed for different languages is shown in Figure 2.8. In the field of MT, more systems have been developed to translate Hindi and English languages as per the analysis presented graphically in this figure. The details of research work for other ancient Indian languages such as Punjabi, Bengali, Marathi, Telugu, Tamil, Assamese, Urdu, Malayalam, Gujarati, Sanskrit, Kannada, Dogri, Sinhala and Devnagari are provided in Table 2.15. It can be observed that work for the Sanskrit language is minimal despite huge corpora and literature.

Table 2.13: Human-Engineered Systems Developed for Various Languages

Author	**Corpus**	**Language Pair**	**Features**	**Outcome**
		Technique:Rule-based		
Adapanawar *et al.*[62]	General Database	English-Marathi	Open source NLP tools are used for developing the system. Database of rules using a bilingual dictionary is used for mapping	The research represents a theoretical and grammatical framework which is extendable.
Adak [70]	Parallel corpus	English-Bengali	Use a soft computational technique where the fuzzy If-Then rule is applied to choose a lemma from prior knowledge.	The proposed system works well in sentence translation from English to Bengali and they obtain 82.92% F-measure based on their test case analysis.
Pisharoty *et al*[63]	General Database	English-Marathi	For improving the performance of the system, grammar and spell checker can be used. The sentiment analysis module can also be used.	The additional functionalities have improved the system accuracy, although there is a trade-off of time.
Garje *et al*[60]	1000 sentences	English-Marathi	Semantic and morphological properties are maintained in the lexicon Grammatical structure of the target language gives importance for better translation	The accuracy of the system using TDIL corpora is 44.29% and for human translation is 49.78%
Basavaraddi *et al*[61]	General Database	English-Kannada	Complex morphology of the target language handled by the morphological generator. Syntax reordering overcomes syntactic differences.	Differences were found in the syntactic module (word order and morphological level)
		Technique: Transfer Based		
Nair and Peter[64]	1000 sentences	Malayalam-English	Artificial intelligence techniques used for system development. Splitter for splitting compound words, bilingual dictionary, the morphological parser	The system was tested for 1000 different sentences and reported true result for the sentences which had two subordinate clauses. The system is easily extendable for other language pairs.

Table 2.14: Machine-Engineered Systems Developed for Various Languages

Author	Corpus	Language Pair	Features	Outcome
		Technique: Statistical Based		
Kumar and Kumar [89]	1000 names	Punjabi-Engish	System has training to learn and transliteration.	The accuracy achieved by the system is 97% Test set gives the BLEU score 32.11.
Ali *et al*	20173 Sentence Pair	English-Urdu	Moses is used for language training with modeling toolkit IRSILM	NA
		Technique: Phrase-Based Statistical		
Pingali and Vasudeva[184]	43,500 sentences	Telugu-English	Modules: language model, translation model, and decoder	The system exhibits based on geometric average fluency of 2.693 and Adequacy of 2.93
		Technique: Technique: Hierarchical phrase-based		
Khan*et al*[87]	EMILLE Corpus (12500 Sentences)	English-Urdu	EMILLE corpus is used. On Urdu monolingual corpus language model is built. Using SRILM toolkit N-gram model is used	BLEU score for the system in 5-fold test data, 40% (Phrase-based) and 29% (Hierarchical-based) NIST score for the system is 73% (Phrase-based) and 63% (Hierarchical-based)
		Technique: Hybrid		
Nithya and Joseph[156]	563 sentences	Malayalam-English	A statistical method is applied to the corpus and applying machine learning techniques for translation.	BLEU score for the baseline system was 68.14 and for the hybrid system was 69.33
Dhore and Dixit [185]	1000 Words	English-Devnagari	The Multilingual dictionary is created using 1000 banking glossary, which is available on the RBI website. C language is used for the lexical analyzer. For running the system Bison tool is used	The multilingual dictionary is improved and semantic rules in the parser design have also improved
Nair et al[64]	General database	English-Hindi	New Rules has been added	The proposed system design shows the accurate results than other systems that are 94%.
Godase and Govilkar*et al*[186]	Dictionary Database	English-Marathi	Bilingual corpus is used for training. Parsing is used by the system	It makes resources available to everyone by presenting a complete architecture and several algorithms for the system
Singla and Baghla[187]	15000 Sentences	English-Punjabi	Using rule-based modelling technique parsing is performed by the system	The accuracy achieved by the system is 81.67%
		Technique: Example Based		
Sinhal and Gupta[188]	677 Sentences	English-Hindi	Comparing sentence to extract the translation. Parallel corpus is used for Training. Uses various modules such as similarity matrix, training matrix, tagging matrix	The system provides 96.07% word strength and 86% precision for 677 sentences
Anuj and Manoj Kumar[189]	General Database	Malayalam-English	Phases of the system are an acquisition, matching, and recombination Searching mechanism is used for searching fragments of Malayalam	Best translation quality is given by the 75% test and reordering problems

Table 2.15: Comparison of Modelling Techniques for Indic Language Pairs

Language Pair	SMT	RBMT	NMT	Hybrid
Bengali-Hindi	✓	✗	✓	✓
Marathi-Hindi	✓	✗	✗	✗
Telugu-Hindi	✓	✗	✗	✗
Tamil-Hindi	✓	✗	✓	✗
English-Hindi	✓	✓	✓	✓
Assamese-English	✓	✗	✗	✗
English-Urdu	✓	✓	✗	✓
Punjabi-English	✓	✓	✗	✗
Bengali-English	✓	✗	✗	✗
Hindi-English	✓	✗	✗	✓
Malayalam -English	✓	✗	✗	✓
Tamil-English	✓	✗	✗	✗
Telugu-English	✓	✗	✗	✗
Urdu-English	✓	✗	✗	✗
Bangla-English	✓	✗	✗	✗
English-Malayalam	✗	✓	✗	✓
English-Tamil	✗	✓	✓	✗
English-Dogri	✗	✓	✗	✗
English-Marathi	✗	✓	✗	✗
English-Bengali	✗	✓	✗	✗
English-Kannada	✗	✓	✗	✗
Punjabi-Hindi	✗	✗	✓	✗
Gujarati-Hindi	✗	✗	✓	✗
Urdu-Hindi	✗	✗	✓	✗
Sinhala-Tamil	✗	✗	✓	✗
Tamil-Sinhala	✗	✗	✓	✗
English-Punjabi	✗	✗	✓	✗
English-Gujarati	✗	✗	✗	✗
English-Devnagari	✗	✗	✗	✗
English-Sanskrit	✗	✓	✗	✓
Sanskrit-Hindi	✓	✗	✗	✗

2.5 Machine Translation System for Processing Sanskrit and Hindi Languages

In a large multilingual society like India, it is of great importance to have a translation system which may convert the different languages into the language of individual interest. Sanskrit is the mother tongue of 24,821 people and Hindi of 52,83,47,193 people, i.e., 43% of total languages in India according to the Census of India[19]. Sanskrit is considered as the donor of almost all Indian languages[196]. The vast reserves in the Sanskrit language can be converted into other languages[21]. The rich knowledge base in Sanskrit is its grammatical tradition attracting Indian and western scholars. It is one of the spoken lan-

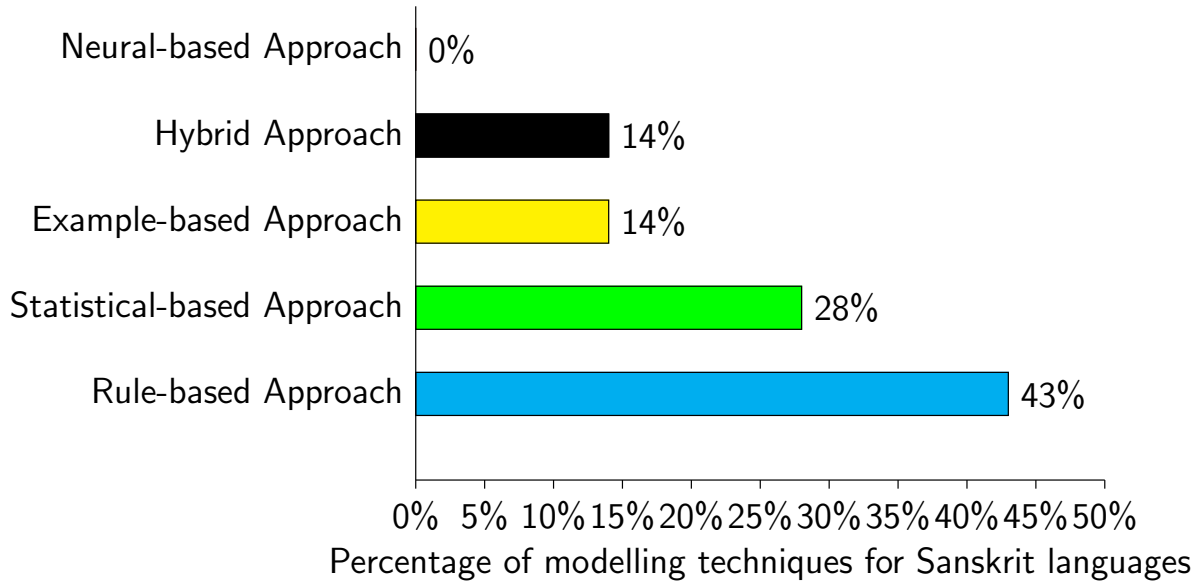


Figure 2.9: Different Modelling Techniques for Sanskrit Language

guages and also at one time was known as 'Lingua Franca' of the world intellectuals[24]. Sanskrit has the text of different domains ranging from Ayurveda, Philosophy and Astronomy. It holds a rich grammar confined by Panini nearly 2500 years ago formulating 3,949 rules which extended later on[2]. Sanskrit has the strongest and simple non-ambiguous grammar[22]. It has the richness of scientific literature with extensiveness and comprehensive analysis, structured approach, and traditional grammar[20]. Many people have attempted to write a grammar for Sanskrit language using the Paninian framework and used it to develop translation system[23]. The Sanskrit grammar is termed as 'Father of Informatics' as it builds a relationship between speech and utterance of speaker and meaning derived by the listener[25]. Hence, the primary objective of Paninian grammar is to form a theory of human natural language communication. Sanskrit and Hindi belong to the same Indo-Aryan family[26]. They both have structural and lexical similarity as Hindi inherits from Sanskrit. Sanskrit has the rich and structured grammar in the form of Panini Astadhayayi, whereas in Hindi such parallel grammar does not exist. Therefore, it becomes difficult to map the divergence between these two languages. The non-existence of such grammar leads to exceptional cases which uncover linguistic generalisations such as Vibhakti in Hindi. Despite rich grammar, choosing Sanskrit as a source language is difficult because parsing fails due to its synthetic nature in which single word can run up to 32 pages. With the rich diversity of grammar, text and resources, it is perplexing to find access to Sanskrit computational tools. One of the many reasons stated is difficulty to access the literature as Sanskrit scholars have not yet turned towards computer science. There are few systems for processing Sanskrit language and translating it to English or vice versa and also Hindi. These are depicted in Table 2.17. There are decidedly fewer systems for translation of Sanskrit to Hindi as compared to English-Sanskrit as

displayed in Figure 2.10. Different modelling techniques used for processing the Sanskrit language are depicted in Figure 2.9. There is a lot of work done for RBMT systems for the Sanskrit language, whereas no work has been done in Sanskrit Translation using Statistical modelling technique. The other techniques like example-based and hybrid-based are equally used for Sanskrit translation. The performance of these techniques for the Sanskrit language is exhibited in Figure 2.10. Hybrid has performed much better than other techniques by achieving more than 37% improvement as compared to others.

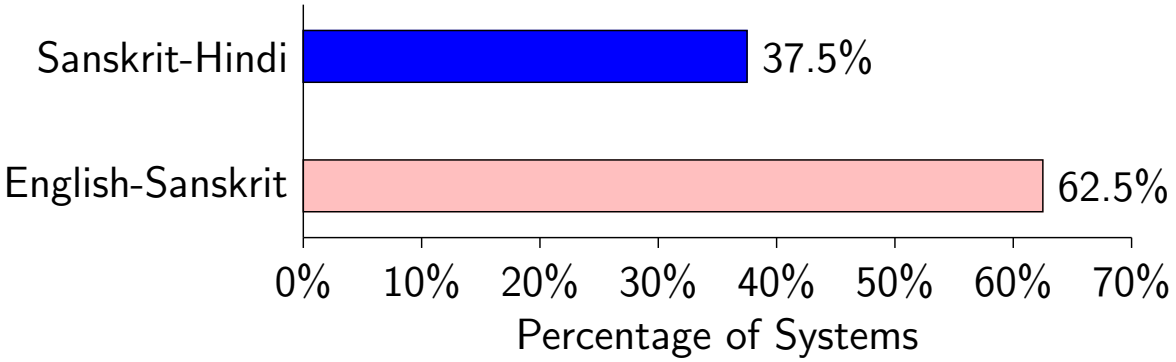


Figure 2.10: Translation Systems for Processing Sanskrit

2.6 Existing Machine Translation System on Cloud

The neural network-based approach is being used prominently in the development of MTS. It has significantly improved the results after statistical MT. These systems deployed on the cloud provide a scalable service with various benefits. Some of the translation systems deployed earlier on the cloud have been discussed in this section. Kiyurkchiev et al.,[197] developed a DisPeL Learning Portal which is an e-learning portal for learners and tutors. It is implemented using a combination of Angular 4 and a proprietary SPA framework based on jQuery and vanilla JavaScript. It provides better scalability and cognitive services like automatic translations and search. The advanced reporting instruments via dedicated reporting databases and PowerBI. The application skeleton is developed in NET Framework 4.6. The system provides a quick implementation of the report generator job as another type of application supported by NET Core. Huang et al. [198] provides a scalable model-parallelism library for training giant networks. The single model has achieved 84.4% accuracy; the top-5 validation accuracy with single-crop is 97%. The Re-computation time-23%, Load-Imbalance-3.2% and minimizes the performance overhead. Chen et al., [199] provide a fully synthesizable C++ template library with the considerations of FPGA implementation. An optimized system configuration is developed to maximize overall performance. The software stack is merged with the DNN accelerator to provide a complete system-level solution for the users who need acceleration services. Vaswani et al., [200] developed a library by using Google Brain and

DeepMind.This library of deep learning models and datasets designed to make deep learning research faster and more accessible Tensor to Tensor uses TensorFlow throughout the implementation. It benefits the researcher to train models on CPU, GPU (single or multiple), and TPU, locally and in the cloud, usually with no or minimal device-specific code or configuration. There have been a strong focus on performance as well as usability. The future work may reduce this scaling factor of encoder and decoder model. Venugopal and Zollman[201] proposed an end-to-end grammar-based Statistical MTS running on Hadoop. This system was able to scale to build grammars for large scale in reasonable time frames. Tordera et al. [202] proposed CloudLM, an open-source cloud-based language model intended for MT with distributed architecture. This proposed system is stable, robust and very efficient. Ahmad et al. [203] proposed a MapReduce-based framework for machine translation which enhanced the throughput of the system. Kumar et al.[204] proposed an MTS using MapReduce Framework to assure QoS. With the experimental analysis, they showed that job completion time for any translation was within a fixed time limit irrespective of its size. There are a few other MTS proposed with Cloud-based framework [[205],[206],[49],[207]]to achieve better performance for current MTS. Another experiment of deployment by Ahmad et al. [208] using Storm, a distributed computing framework was used for deploying on cloud. It reduced job completion time, gave very good user experience by web interface and response time. The elapsed time for a translation job after its submission was 4 seconds. In a future modification of the storm topology to apply parallelism at the word level processing, different MT systems developed on the cloud were compared based on techniques used, technical specifications employed, and the outcome achieved with the limitations and future work. A comparative analysis of various cloud-based approaches for MT is presented in Table 2.18 and 2.19.

2.7 Conclusion

This Chapter, reviews the different modelling techniques and the resources required for modelling the techniques such as corpus, domains, toolkits, models, features and their evaluation measures.It further contributes to the research community by providing a comparative analysis of research work on different Indic language pairs based on modelling techniques. The synthesis from the comparative analysis concludes that the work on Sanskrit-Hindi language pair is minimal, despite holding an ancient scientific and comprehensive literature. As per the analysis, the translation systems for translating any language to English and Hindi are greater in number than other languages. The literature survey leads us to conclude that the use of SMT-based MTS is more, i.e. 41%, as compared to others, whereas the use of hybrid MTS is 14%; rule-based is 22%; neural-based is 23%. Further, neural and hybrid approaches perform better as compared to other techniques. Thus, these techniques may be considered for future use. RBMT and

SMT are the approaches most frequently used, but their accuracy is low. Therefore, more accurate and efficient techniques i.e., NMT and HBMT are required to be implemented.

The next chapter presents the developed Sanskrit-Hindi hybrid MTS. It layout the description of the system by extracting features from rule-based and feeding them to the neural network to form a hybrid system for translation from Sanskrit to Hindi.

Table 2.17: Comparison of Sanskrit and Hindi Language Processing Systems

Author	Technique	Language Pair	Corpus	Features	Results	Issues
Pandey and Jha[190]	Statistical-based	Sanskrit-Hindi	24,000 (bilingual) 25,000 (monolingual)	The System is being trained simultaneously on Microsoft Translator Hub (MTHub) and is intended only for simple Sanskrit prose texts.	39.17 % for long, complex and compound sentences. 41.17% for bilingual and monolingual sentences.	The system takes input only in Devanagari Unicode script and gives output in same. Sometimes, the system does not respond to long and compound sentences.
Rathod et al[191]	Rule and Example-based	English-Sanskrit	NA	The text input is processed with spell-checker, followed by token generator, translator, parser, EBMT/RBMT database and generator.	An average improvement of 10% has been achieved by using EBMT in the translation quality which is more than that of RBMT	EBMT is better than RBMT, but the performance degraded in the case of extra large sentences.
Shukla and Shukla[192]	Rule-based	English-Sanskrit	NA	It unifies the isolated word class under the Speech Recognition type, traditional dictionary rule-based machine translation technique and text to speech synthesizer.	An average improvement of 7% has been achieved in the translation quality.	Translates only simple sentences, not complex sentences.
Bahadur et al.[193]	Rule-based	English-Sanskrit	500 sentences	The sentences can be simple and compound with the affirmative and imperative type or of active or passive voice having any of the three tenses, i.e. Present, Past, and Future	An average improvement of 13% in the translation quality, and 90% accuracy for extra learge sentences has been achieved by this modelling technique.	The sentence is correct in terms of grammar, but the translation is not correct. Few words in English may be used as both noun and verb. This generates ambiguity for the system.
Jayan et al.[194]	Example-based	English-Sanskrit	125 input-output pairs	Proposed a novel method that uses rules and ANN technique to detect and implement the adaptation rules for the divergence in English to Sanskrit machine translation.	An average improvement of 8% has been achieved in the translation quality.	For better improvement, future work is carrying to perform case-based reasoning in a combination with rule-based and ANN model for this purpose.
Subramanian et al.[195]	Rule-based	English-Sanskrit	NA	Since the conjunction of individual words of Sanskrit into longer ones is a common occurrence in Sanskrit texts, sandhi vichcheda is the most important part of the translator	An average improvement of 7% has been achieved in the translation quality.	Semantic ambiguities were not handled and pragmatics and discourse considerations were out of the scope. Some among them are the ambiguities of use of right prepositions and plurals of words. Another aspect that has not been covered is structuring of the sentence.

Table 2.18: Existing Work of Machine Translation Deployed on Cloud

Research Work	Year	Research Organization	Stream	Methodology	Results	Future Research Directions
[197]	2019	Plovdiv University	Statistical+Neural	DisPeL Learning Portal is an e-learning portal for learners and tutors. Implemented using a combination of Angular 4 and a proprietary SPA framework based on jQuery and vanilla JavaScript	Better scalability DisPeL with cognitive services like automatic translations and searching Advanced reporting instruments via dedicated reporting databases and PowerBI.	The application skeleton in NET Framework 4.6. Quick implementation of report generator job as another type of application, supported by NET Core
[198]	2019	NA	Neural Network	A scalable model-parallelism library for training giant networks.	Single model achieved 84.4% accuracy Top-5 validation accuracy with single-crop is 97%.	Re-computation time-23%, Load-Imbalance-3.2% and minimize the performance overhead.
[199]	2019	University of Illinois at Urbana-Champaign,USA.	Neural Network	A fully synthesizable C++ template library with the considerations of FPGA implementation.	An optimized system configuration to maximize overall performance. Software stack together with the DNN accelerator, to provide a complete system-level solution for the users who need acceleration services.	NA
[200]	2018	Google Brain and DeepMind	Deep neural network	It has a library of deep learning models and datasets designed to make deep learning research faster and more accessible T2T uses TensorFlow throughout.	Researchers can train models on CPU, GPU (single or multiple), and TPU, locally and in the cloud, usually with no or minimal device-specific code or configuration. A strong focus on performance as well as usability.	Future work may reduce this scaling factor of encoder and decoder model.
[202]	2016	ADAPT Centre,School of Computing ,Dublin City University,Ireland	Statistical	Language Model(LM) in Apache Solr. Cloud-based LM is in Moses. Efficiency enhancement by cache and block query.	Cache reduces 70% memory usage. Memory used argument by 20% Sentences decoding 100 sentences reduce the time by using cache 89% and memory used increments by 195%. Block queries when decoding 1 sentences(9% faster)and slower for 10 to 100 sentences.	Enhancing the efficiency and time by keeping the connection alive between the Moses and Solr (so that a new query does not need to reopen the connection)
[208]	2014	LTRC, IIIT Hyderabad	Statistical	Storm, a distributed computing framework is used for deploying on the cloud.	It reduces the job completion time. It gives very good the user experience by web interface Response time. The elapsed time for translation job after the submission of the job is 4 sec.	In the future modification of the storm topology to apply parallelism at word level processing.
[49]	2013	IIIT Hyderabad,India	Transfer	CentOS 5.3 as guest O.S. Xen is used for virtualization of hardware. 5.7 as Host operating system. Hadoop as middleware for word load partitioning. Eucalyptus for setting up the cloud infrastructure.	To translate a book in the stand alone system took 71 minutes and 25 seconds and on eucalyptus, cloud environment took 5 minutes and 27 seconds. It requires 3 minutes to scale up the MT application after provisioning the new resources in the system.	Enhancing the virtual appliance so that it is available as repositories and from there can be deployed in the cloud that would enable the MT system to handle frequent updates as well.
[206]	2013	IIIT Hyderabad, India	Transfer-based	Sampark machine translation system is used. Hadoop, open-source implementation of MapReduce for partitioning jobs. Eucalyptus cloud for running the cloud system. For virtualization CentOS(5.7)as host O.S with Xen.	A cluster of 5 nodes for 25600 sentences total time to compute is 12920, and throughput is 119 per minute. If the partition size is doubled the throughput increases by less than 5%.	Extension of the approach for QoS to various NLP applications that exhibit list homomorphism and can be partitioned on distributed computing.
[209]	2012	University of Edinburgh, Zagreb, Copenhagen, Uppsala, Moravia	Statistical	MT training and decoding tool Moses is used.	Achieved better quality,reputation,convenience and domain-specific translation for user-specific text. Ready resource for studying and teaching purpose for academic institutions.	Support for only some specific format is provided in this version of the system. For scalability, the system depends on cluster hosted on Amazon Web service infrastructure.
[207]	2011	Moravia, Semlab and University of Edinburgh	Transfer	SMT training and decoding toolkit on Moses. Moravia used LetsMT platform to train and evaluate SMT systems for polish and Czech.	The system has to lead a strong increase in translator productivity.	Depended on translation memories for parallel data. Translators receive translation suggestions provided by the selected MT engine running on LetMT!.
[210]	2010	Carnegie Mellon University, United States	Phrase-based	Chaski toolkit is run on the top of the Hadoop framework. GIZA++ and Moses are used for training data and aligning data.	The training time of word alignment is reduced from 47 hours to 8 hours. Time for phrase extraction from 21 hours to 43 minutes. The output phrase table is compatible with the Moses decoder.	Distributed word alignment technique used has no changes on performance. Chaski toolkit used and Moses phrase table has no significant difference. Language model focuses on the validity of results rather BLEU score.
[201]	2009	Carnegie Mellon University, United States	Statistical	Syntax Augmented Machine Translation(SAMT) on map-reduce Hadoop for parse tree of the target language	Translation quality measured with IBM-BLEU % for 67M it took 25.88 min and for 230M it took 26.28 minutes.	Using SAMT(Syntax-based Augmented Grammar), the system exhibits consistency but with small improvements.

Table 2.19: Existing Commercial MTS:A Comparative Study

Commercial System	**Developer**	**Platform**	**Language Pair**	**Approach**	**Findings**
Bing Translator	Microsoft	Web application	52	Statistical	It is a web service with back-end translation software. For translating an entire web page 4 bilingual viewer layouts are present.
Microsoft Translator	Microsoft	Windows,Apple and Android phone	60 (for text) and 9(for speech)	Statistical	Available for both personal use and business use.
IBM Translator	IBM	Web service	5 (conversational domain),10 (news domain),5 (Patent domain)	Hybrid	It is a translation service with multi-domain translations in real time.
SDL Language Weaver	SDL Trados,Germany	Web application	111+	Statistical	It is a computer-assisted translation translation software suit.
Slate	Precision Translation Tools	Windows,Linux(i86-64)	29	Statistical	It is a personalized translation engine outperforming cloud-based MT without their drawbacks requiring less amount of time.
Kantan MT	Xcelerator Machine Translation Ltd.	Web application, Windows,MAC, Linux	59	Statistical	A high quality MT service is provided using the power and flexibility of cloud.
Xerox Easy Translator	Xerox Easy Translator Service	Web application	57	Hybrid	It offers professional translation service through multiple access points,offering flexible, secure and affordable services for translation.

Chapter 3

Proposed Sanskrit-Hindi Hybrid Machine Translation System

The previous chapter primarily aims at reviewing the existing literature on the subject under study i.e., Machine Translation(MT). It was a modest endeavour to study in detail the extant modelling techniques of MT. It provided a chance to deeply understand the various issues and aspects of the current study. It also served the purpose of finding the gap in the research area and avoid duplication.

This current chapter presents the proposed system that overcomes earlier works for translating Sanskrit-Hindi which lack extensibility, generalizability and adaptability. This proposed system extracts features from the rule-based system as linguistic rules and feeds them further to train the recurrent neural network. This work is novel and applicable to any low-resource language pair with minimum linguistic knowledge.

The chapter organised as follows: Section 3.1 presents the system description of our proposed work. The experimental design presented in Section 3.2 while Section 3.3 shows the performance evaluation followed by conclusion in Section 3.4.

3.1 Proposed MTS Description

This section describes data pre-processing, addition of linguistic features, employing encoder for embedding source input sentence into vectors, decoder for converting the trained vectors into target sentence.Further the developed translation is made accessible through a web interface for users through the proposed MTS.

3.1.1 Data Pre-processing of Corpus

The Neural Machine Translation(NMT) has been gaining a significant interest of researchers and prominently achieving better results compared to earlier approaches to MT. To meet the requirement of parallel corpora for training an NMT model, gathered a parallel corpus of 90-100k sentences. The corpus data is firstly converted to a suitable format for applying to the NMT model. It is a two-fold process, i.e. clean text and split text. The cleaning of the text involves dividing the document into sentences. Then removing all non-printable characters, punctuation markers, normalise Unicode characters to ASCII value, changing upper case letters to lower case, removing any remaining tokens that are not alphabetic or numeric. The current work performs these operations on each phrase for each pair of dataset loaded. Secondly, the splitting operation has been performed on cleaned data. The dataset contained different length sentence pairs, therefore, different computation graphs. The sentences were sorted in a batch based on sentence pair by length and broke similar-length sentences into mini-batches. Thus, the training corpus was recurrently shuffled and broke the corpus into maxi-batches and again splitting in mini-batches. These were processed by applying gradient for parameter update.

3.1.2 Rule-based Machine Translation System for Extracting Linguistic Features

The proposed work has a pipeline architecture, which takes input from its previous phase and performs computations and passes the output to the next step. Different tools divided into different modules or phases developed under Sanskrit Consortium Project funded by MIT using Anusaaraka[211].There are 10 modules in the rule-based pipeline architecture of Sanskrit to Hindi translation[212]. All of the modules provide an individual output as linguistic features to Neural based Encoder-Decoder to train the system more efficiently. Engine[211] for translation.

- Pre-processing of user input: It takes input from the user, cleanses, normalise the text, converts the input notations into WX notation, call and invokes MT system which performs computation and shows the output result.

- Tokenizer: Tokenizer receives a flow of character, and that character breaks into individual words called as tokens (words, punctuation, markers). It removes the formatting information and adds a sentence tag. Here the term morphology is used for linguistics. It refers to a study of words, their internal process and their word meaning. The model has a stream of words; those words tokenised first, and then morphology gives meaning to those words[213].

- Sandhi-Splitter: It invoked when the input text contains Sanskrit sandhi words. It splits these words as well as compound words[214][215][216].

- Morphological Analyzer: It split words into their roots and grammatical suffixes. There are different units, and each unit provides meaning as well as grammatical function. It also provides inflectional analysis, prunes the answer, uses local morph analysis to handle unrecognised words and produce a derivational analysis of the derived roots.[217][218][219].

- Parsing: Parser used as compiler and interpreter that breaks data into smaller units for easy translation of one language to another. Parsers take input from the sequence of words or tokens. These inputs translated in the form of a parse tree. It converts the source language into target language in the form of a tree with labels of noun, verbs and their associated attributes. Morph analysis according to context along with karaka analysis is performed. According to computational Paninian grammar, it identifies and names the relation between the verb and its participants. [220][221][222][223][224].

- Shallow Parsing: If the parser fails on any input it does minimum parsing of the sentence and produces pruned morph analysis to next layer.[225][215][220][220].

- Word Sense Disambiguation(WSD): The modules perform word sense disambiguation of input sentence words roots, vibhakti, and lakara. It identifies a correct sense of a Sanskrit word[24].

- Parts of Speech Tag(POS): It adds parts of speech tags to each word such as adjective, verb or noun. tags[226][227].

- Chunker: This phase performs a minimum grouping of words in a sentence such as a noun phrase, verb phrase, adjective phrase. The rule base allocates an appropriate chunk tag to it. [228].

- Hindi Lexical Transfer: The Sanskrit Lexicon is transferred to Hindi identifying root words using the dictionary. The output formatted according to the Hindi Generator, which generates the output in Hindi Language corresponding to the Sanskrit language. This module also performs transliteration in case of translation fails[229].

- Hindi Generator: This phase involves sentence level generator which performs agreement between noun, adjective and verb in the target language. Addition of vibhakti markers 'ne' and dropping 'ko' at required positions. Final generation involves root words and their associated grammatical features, corresponding suffixes and concatenates them by generating words into a sentence[24].

Hence, a translation of each Sanskrit word to its corresponding Hindi word is performed as per the linguistic rules and tools. Further, this data is passed on to the consequent phase. Then, it is converted into Comma-separated values(CSV) format suitable for training, model development and fitting the values for Neural-based Encoder-Decoder architecture for predicting translation of Sanskrit word to Hindi word. These linguistic tool's output is embedded as features for input encoding of the source sentence as they help greatly in the disambiguation of words.

3.1.3 SHH-MTS: Neural Network-based RNN Approach

The Sanskrit-Hindi Hybrid Machine Translation System(SHH-MTS) proposed and presented in this section. The previous section describes the extraction of linguistic features from the pipeline architecture. This section describes the feeding mechanism of extracted features into the neural network. In this work, employement of a recurrent neural network with Gated Recurrent Unit(GRU) is used. The encoder-decoder recurrent neural network consists of encoder reading a variable-length input sequence and decoder predicting a variable-length output sequence. The dense output layer is used to predict each character in the output sequence in one time rather recursively during training. Firstly, defining a model; and once it fits, it can be used to make translation predictions. The model defined for training has learned weights for this operation, but the structure of the model is not designed for calling recursively to be generated one character at a time as presented in Figure 3.1. The encoder model takes an input layer from the encoder in the trained model, and gives the output as a hidden layer and cell state tensors. On the other hand, the decoder needs hidden layer and cell state from encoder as an initial state for the model defined. Both the encoder and decoder will be called recursively for each character that is to be generated in the translation sequence[107].
The Neural Network Encoder-Decoder architecture with bidirectional RNN[8] is implemented for predicting Hindi translation corresponding to Sanskrit translation. It consists of Gated Recurrent Units(GRU) for computation. The implementation consists of input sequence fed with linguistic features. Given a source sentence $x = (x_1, x_2....x_m)$ is read, and computes hidden states for forward direction ($\vec{h}_1$, $\vec{h}_2$, $\vec{h}_3$.....$\vec{h}_n$)and for backward states($\overleftarrow{h}_1$,$\overleftarrow{h}_2$.....$\overleftarrow{h}_n$). The detailed computations have been displayed in Algorithm 3.1 . Both of these forward and backward computations are then merged to form an annotation vector(h_i) as explained in Algorithm 3.2. The encoder input was a combination

of linguistic features formed as feature embedding matrices as computed in Algorithm 3.3. The decoder further predicts the target sequence $y = (y_1, y_2, ...y_n)$ based on context vector(c_i) computed in Algorithm 3.4 from weighted sum of annotations h_i, recurrent hidden state (s_i) and previously computed word y_{i-1}.The alignment model (a_{ij}) models the probability that x_j is aligned to y_i or not as in Algorithm 3.5. It is feed-forward single layer network learned through back-propagation. The output is predicted using learned distribution. The implementation is carried out with tensorflow[230] at back end with keras[231] using Encoder-Decoder architecture for developing the system. In addition, consideration of the state contraints in order to minimize the computational complexity, and generalized the results inspired by the work [232].

$$d_{i,iz} = f_i(w_s) + F_{i,z}(W_s, \bar{W_s - 1}) \tag{3.1}$$

where $i = 1...n$, $ji = 1....s_i - 1$, $s_i > 1$,$n > 1$; and both s_i, n are positive integers.

$$d_{i,ji} = d_{i,ji+1} + F_{i,ji}(\bar{d}), ji \tag{3.2}$$

here, $d = [d_i, ji....\bar{d_{i,ji}}]^T$ represents the states of encoder and decoder mechanism. The $\bar{d_{i,ji}} = [d_{i1}...d_{i,ji}]^T \in R_{k_s}$

$$W_o = d_i, 1 \tag{3.3}$$

where W_o denotes the output of system. $W_s =\in \bar{R}(\bar{Ws - 1} = [W_s...Ws - 1]^T)$ denotes the system input. $f_i(W_s)$ denotes the hysteresis type of non-linearity. $F_{i,ji}(\bar{W}_s, ji)$ denotes the smoth function. For the proposed system, the state constraints are along with $-Q_{i,ji} < d_{i,ji} < \bar{Q_i}, ji$ which denotes the positive design constraints, where, $i = 1....n, ji = 1....s_i$. The consideration of these state constraint for the propsoed system resulted in its stability. Assuming μ_i,ϕ_i,ψ_i= design parameters; where, $\phi_i > 0$ are slopes of lines and $\phi_i > \psi_i$. It will exhibit the change in equation(1), Here, the states are modified after the constraint imposition as $f_i(w_s)$=$\phi_i(W_i(t))$+ $\rho_i(W_{s_i})$

$$d_{i,iz} = f_i(w_s) = \phi_i(W_i(t)) + \rho_i(W_{s_i}) + F_{i,z}(W_s, \bar{W_s - 1}) \tag{3.4}$$

where, $\rho_i(Ws_i) = bounded$ and satisfied. All other parameters of the system will remain the same, and the proposed system satifies the state contraint for adaptive neural network used for translating a sentence from source to target language.The developed system exhibits BLEU, i.e., an automatic measure for translation accuracy as 61.02% on combining Keras model with bi-directional layer using gated recurrent units along with Relu and sigmoid activation function, and then performing the auto-tuning. Adam optimizer is also used to optimize the neural model. The step-wise details have been given in further

subsections.

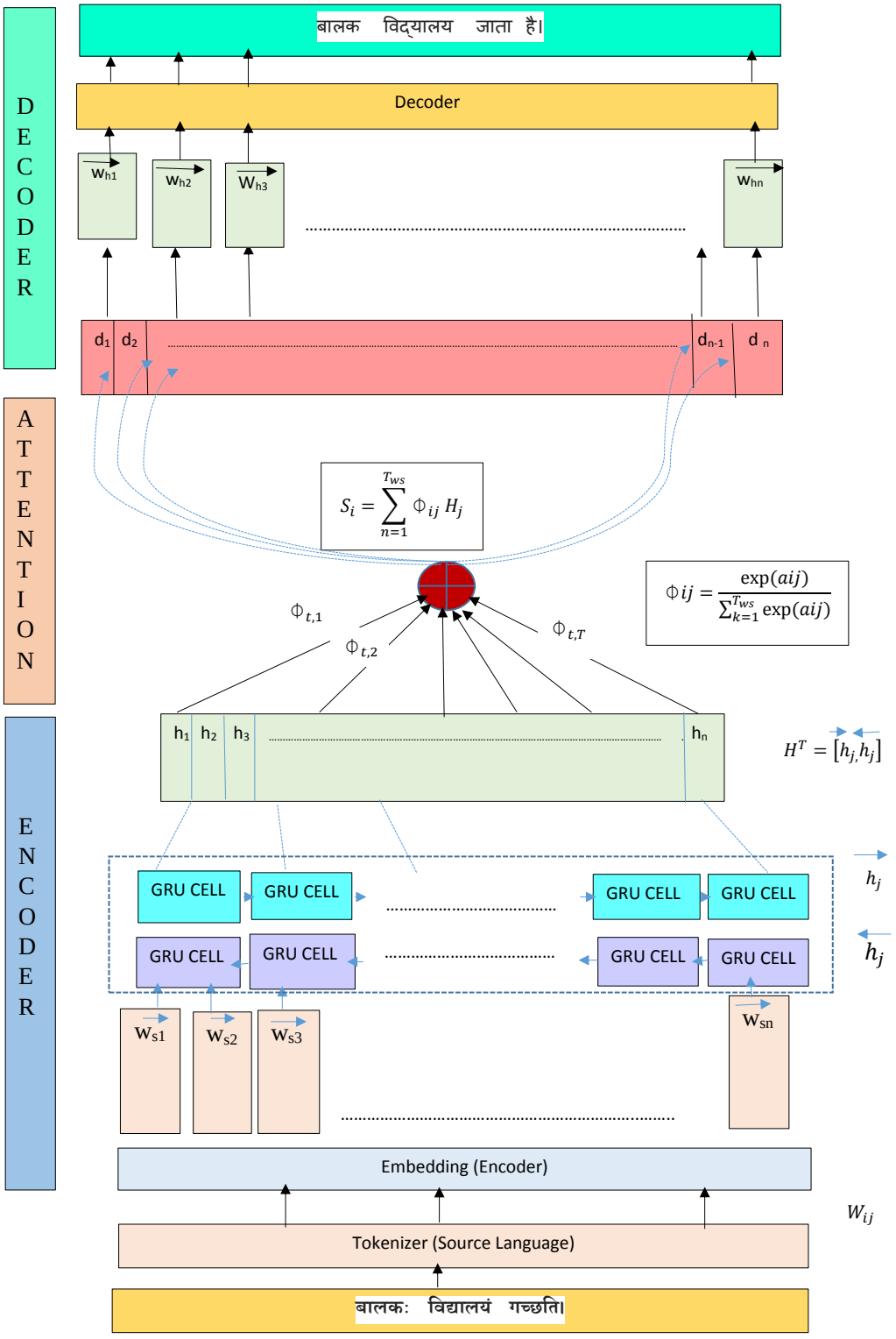


Figure 3.1: Deep Neural Network Architecture

3.1.3.1 Encoder

Given a source sentence in Sanskrit Language $W_s = w_{s_1}, w_{s_2}, w_{s_3}..w_{s_z}, s_i \in \mathbb{R}^{Ks}$ and target sentence in Hindi Language $W_h = w_{h_1}, w_{h_2}, w_{h_3}, ...w_{h_l}, h_x \in \mathbb{R}^{kh}$ from the parallel corpus. Here, K_s and k_h are vocabulary sizes; and z and x are length of input sentence and output sentence. The model first tokenizes W_s to form input representation, where probability of a sequence of $T(w_{s_1}, w_{s_2}, ...w_{s_n})$ words is denoted as $P_1(w_{s_1}.....w_{s_t})$. It is usually conditioned on window of words rather than all previous words. Since the number of words coming before a previous word w1 varies depending on locations with input document in Eq.(3.5).

$$P_1(w_{s_1}, w_{s_2}....., w_{s_{t_z}}) = \prod_{i=1}^{t} P(w_{s_1}.....w_{s_{i-1}}) \approx \prod_{i=1}^{t} P(w_{s_i}|w_{s_1}....w_{s_{z-1}})....w_{s_{z-1}} \tag{3.5}$$

As direct application of neural network to text data cannot be applied. Text is firstly converted into numbers or integer-tokens which are further converted into vectors by embedding layers. By setting the maximum number of words in the vocabulary, tokenizer for source and target language is used. The dataset once converted into sequence of integers-tokens are then padded and truncated, and saved as numpy arrays. The Encoder uses this output of tokenizer as arrays, and computes embedded vectors $(\vec{w_{s_1}}, \vec{w_{s_2}}, \vec{w_{s_3}}.....\vec{w_{s_z}})$ for hidden layers computation. These vectors have values ranging between 1 and -1 having similar semantic meaning words mapped to similar vectors.

Forward RNN reads input sentence from starting to end $\vec{f}$ and compute the hidden states $(\vec{h_1}, \vec{h_2}, \vec{h_3}..\vec{h_{\tau_j}})$. The backward RNN computes the hidden states$(\overleftarrow{h_1}, \overleftarrow{h_2}..., \overleftarrow{h_{\tau_j}})$ by reading the sentence in a reverse order. These hidden states, i.e., forward and backward are combined to form an annotation vector $H_i = [\vec{h_j}^T; \overleftarrow{h_j}^T]$. The traditional encoder consisting of an embedding lookup of each input word s_z and mapping steps through hidden states $\vec{h_\tau}$and $\overleftarrow{h_\tau}$in Eq.(3.6).

$$\vec{H_j} = f(h_i \vec{-} 1, \bar{E}W_{s_n}) \tag{3.6}$$

The encoder computations are deeply stacked in the following manner as in Equations (3.6 and 3.7). For the first layer,

$$h_{t,1} = f_1(h_{t-1}, 1, w_s t) \tag{3.7}$$

For $i > 1$

$$h_{t,i} = f_{h_{t-1,i}, h_{t,i-1}} \tag{3.8}$$

where,

$h_{t-1,i}$: stands for previous time stamp value and $h_{t,i-1}$: for previous layer in sequence value.

The context vector s_i contains the summary of the input sentence computed by processing backwards and forward RNN's. For the proposed system, gated recurrent units have been used for the function of encoder as well as decoder[233]. The Gated Recurrent Units(GRU)'s have been designed in a manner to have more persistent memory, thereby making it easier for RNN to capture a long-term dependency. Mathematically, GRU has previous state h_{t-1} and input w_{s_t} to generate the next hidden state h_t.
For update gate in Eq.(3.7), reset in Eq.(3.8), new memory in Eq.(3.9), and hidden state for all I words of a sentence in Eq.(3.10).

$$\vec{up}_i = \sigma(\vec{W}_{up}\bar{E}_{s_i} + \vec{O}_{up}\vec{h_{i-1}}) \tag{3.9}$$

$$\vec{res}_i = \sigma(\vec{W}_{res}\bar{E}s_i + \vec{O}_{res}\vec{h_{i-1}}) \tag{3.10}$$

$$\vec{h}_i = tanh(\vec{W}\bar{E}s_i + \vec{O}[\vec{res}_i \odot \vec{h_{i-1}}] \tag{3.11}$$

$$h_i = (1 - \vec{up}_i) \odot \vec{h_{i-1}} + up_i \odot \vec{h_i}) \tag{3.12}$$

Here, d is the dimensionality of word embedding and u is number of hidden units.

$$\bar{E} \in \mathbb{R}^{d \times ks}.$$
$$\vec{W}, \vec{W}_{up}, \vec{W}_{res} \in \mathbb{R}^{u \times d}$$
$$\vec{O}, \vec{O}_{up}, \vec{O}_{res} \in \mathbb{R}^{u \times u}$$

σ is logistic sigmoid function. The backward states of bidirectional recurrent Neural network are computed similarly for update gate in Eq.(3.11), reset gate in Eq.(3.12), new memory in Eq.(3.13), and hidden state for all i words of a sentence in Eq.(3.14).

$$\overleftarrow{up}_i = \sigma(\overleftarrow{W}_{up}\bar{E}_{s_i} + \overleftarrow{O}_{up}\overleftarrow{h_{i-1}}) \tag{3.13}$$

$$\overleftarrow{res}_i = \sigma(\overleftarrow{W}_{res}\bar{E}s_i + \overleftarrow{O}_{res}\overleftarrow{h_{i-1}}) \tag{3.14}$$

$$\overleftarrow{h}_i = tanh(\overleftarrow{W}\bar{E}s_i + \overleftarrow{O}[\overleftarrow{res}_i \odot h - \overleftarrow{i} - 1] \tag{3.15}$$

$$h_i = (1 - \overleftarrow{up}_i) \odot \overleftarrow{h_{i-1}} + up_i \odot \overleftarrow{h}_i) \tag{3.16}$$

The forward and backward states are combined as $h_i = [\vec{h_i} + \overleftarrow{h_i}]$

3.1.3.2 Addition of Linguistic Features to Encoder

The framework of the current research integrates linguistic features[234] extracted from the pipeline architecture of rule-based to train Recurrent Neural Network. Each feature has a distinct vector word embedding s_{zy}. Combining all these word vectors form a feature embedding matrix $E \in \mathbb{R}^{dy \times ky}$ with d_k as a summation of the dimension of all feature embeddings and ky as vocabulary size of K^{th} feature.These embeddings are later concatenated with total embedding size as the length matches. The input embedded sentence vectors are multiplied with these extracted linguistic features. All other functionality and parameters of the model remain the same, only this change in the encoder is performed as in Eq.(3.15) which result in exceptional improvement in the fluency of output.

$$h_l = tanh(\vec{W} \prod_{y}^{F} \bar{E}_y s_{zy} + \vec{O}\vec{h}_{l-1}) \tag{3.17}$$

3.1.3.3 Attention Mechanism

The attention layer bridges the gap of Encoder that produces a sequence of word representation $h_j = (\vec{h_i}, \overleftarrow{h_i})$ and decoder expecting S_i context vector at each time step t_i. It calculates the association between input word W_s to produce the next output word w_h by calculating the impact of word representation $(\vec{h_i}, \overleftarrow{h_i})$. The context vector is mathematically calculated as a weighted sum of annotations h_i. For this, first need to calculate alignment model a_{ij} as in Equations(3.16-3.18), the score of output position around i to input position around j. It takes hidden state d_{i-1} and h_j as j^{th} annotation of input Sanskrit sentence.

$$a_{ij} = J_a^{\tau} tanh(W_a d_{i-1} + O_a h_j) \tag{3.18}$$

$$\alpha_{ij} = \frac{exp(a_{ij})}{\sum_{y=1}^{Ts} exp(a_{iy})} \tag{3.19}$$

$$S_i = \sum_{j=1}^{ts} \alpha_{ij} h_j \tag{3.20}$$

Here, S is feed-forward Neural Network.
$W_a \in \mathbb{R}^{n'_1}, O_a \in \mathbb{R}^{n' \times n}, J_a \in \mathbb{R}_{n' \times 2n}$ are weight matrices. The computed scalar attention value is normalized using softmax activation function, so all input words adds up to 1.

3.1.3.4 Decoder

The decoder at each time step t, takes sequence of previous hidden state d_{i-1}, some representation of input context s_i and embedding of previous word output $E_{h_{i-1}}$ to output a new word prediction w_{h_i} and new output decoder hidden state. The initial hidden state is computed in Eq.(3.19).

$$d_0 = f(W_d \vec{h_1}) \tag{3.21}$$

where $W_d \in \mathbb{R}^{d \times d}$. The hidden state d_i is computed given annotation from encoder in Eq.(3.20) and for update in Eq.(3.21), reset in Eq.(3.22).

$$d_i = tanh\bar{(}WEh_{y_{i-1}}) + O[res_i + d_{i-1}] + Ss_i) \tag{3.22}$$

$$up_i = \sigma(W_{up}E_{h_i-1} + O_{up}d_{i-1}S_{up}s_i) \tag{3.23}$$

$$res_i = \sigma(W_{res}Eh_{i-1} + O_{res}h_{i-1} + S_{res}S_i) \tag{3.24}$$

Where, E is embedded matrix of word for target language with u as number of hidden units and d is word embedding dimension. $W, W_{up}, W_{res} \in \mathbb{R}^{u \times d}$,
$0, O_{up}, O_{res} \in \mathbb{R}^{d \times 2d}$ are weight matrices. The vector for prediction p_i for a output word is based on decoder hidden state d_{i-1}, input context s_i and embedding of previous output word h_{i-1} as in Eq.(3.23).

$$p_i = softmax(O_{ot_{d_i-1}} + V_{ot}E_{h_i-1} + S0s_i) \tag{3.25}$$

Where, $V_{ot} \in \mathbb{R}^{2l \times d}, O_{ot}\mathbb{R}^{2l \times u}, C_o\mathbb{R} \in 2l \times 2u$ are output word embedding matrices. On, $E_{W_{h_i-1}}$ condition is repeated and use d_{i-1} rather than d_i as it fragments the encoder state progress from d_{i-1} to d_i for prediction of output word p_i in Eq.(3.24). Here, a token for output word w_{h_i} is the highest value in the vector.

$$p_i = [maxp_i, 2\bar{j} - 1, i, \bar{2}j]^{\tau}_{j=1\ldots,l}. \tag{3.26}$$

Even training is performed accordingly as the network being aware of correct output w_{h_i} assigned a larger probability value as in Eq.(3.25).

$$prob(h_i|d_{i-1}, s_i) \propto (h^{\tau}W_o pi) \tag{3.27}$$

Activation function softmax is used to convert the raw vector into a probability distribution having a sum of the values as 1. Relu[235] has also been used here that combines input to yield the next hidden state. To predict the target variable more efficiently, activation function is passed to the model. It also works as a rectifier. The model follows a deep output as suggested by GF Montufar et al. [236].

3.1.4 SHH-MTS: Hybrid Approach

The hybrid approach uses extraction of linguistic feature, and RNN to translate the Sanskrit language to the Hindi language. In this translation model, the source language is translated to the target language having a lexical gap. It is the process which undertakes a deep analysis of the source language and then its lexical transfer to the target language. Accuracy of the hybrid approach is more than that of the rule-based approach or phrase-based approach. The SHH-MTS is capable of merging the best of both the approaches as it merges linguistic rich features with prominent deep learning approach to provide Sanskrit to Hindi translation. The flow of this system is depicted through Figure 3.2.

3.2 Experimental Design

The section describes corpora, model size, parameter initialisation, and training performed. The experiment performed on different epochs, beams sizes and different sentence lengths which lead to change in the update, BLEU score, training probability and development probability have been discussed in detail in Chapter 5 of the work.

3.2.1 Corpora

The open-source manual corpora of Sanskrit was available as in Table 3.2 while parallel corpora of Sanskrit-Hindi was minimal as shown in Table 3.1 of different domains[237]. On request, a parallel corpus of 50,000 was available from Indian Languages Corpora Initiative (ILCI) [190]. It was not adequate to train the NMT system with this existing corpus. To overcome this difficulty, firstly a parallel corpus of Bhagavad-Geeta was manually curated as shown in Table 3.2. Secondly, a synthetic parallel corpus curated with technique suggested by[238]. Sanskrit prose sentences[239] were also available from JNU. Various other Sanskrit books were available from "Development of Sanskrit Computational Tools and Sanskrit-Hindi Machine Translation System (2008-2012)", funded by DeiTy, Government of India developed under the TDIL program [240]. Entire data was pre-processed, and divided into training, development and test set as shown in Table 3.3.

Algorithm 1: The Encoding of Input Sentence

input : Parallel corpus P_c and Monolingual corpus M_c,$varaince = 0.01$, $mean = 0$,$V_\alpha = 0$

output: Context Vector s_i as summary of input sentence

Encode $W_s = w_{s_1}, w_{s_2}, w_{s_3}.....w_{s_z}$ *as* $s_i \in \mathbb{R}^{Ks}$;
// K_s is vocabulary size and z is input sentence length.
Encode $W_h = w_{h_1}, w_{h_2}, w_{h_3}, ...w_{h_x}$ *as* $h_i \in \mathbb{R}^{k_h}$;
// K_h is vocabulary; and x is the output sentence length.
for $s = 1$, $s{+}{+}$, *while* $s < z$ **do**
    Tokenize the input sentence$T(w_{s_1}, w_{s_2}, ...w_{s_n})$;
    Compute probability
    $P_1(w_{s_1}, w_{s_2}....., w_{s_{t_z}}) = \prod_{i=1}^{t} P(w_{s_1}.....w_{s_{i-1}}) \approx \prod_{i=1}^{t} P(w_{s_i}|w_{s_1}....w_{s_{z-1}})....w_{s_{z-1}}$;
    // probability of a sequence of token words conditioned on window words rather than all previous words
The dataset converted into sequence of integers-tokens$(T(w(s_1)...w(s_(z))))$ *are then padded and truncated and saved as numpy arrays(np).*;
for *np.w(s)* **do**
    Apply Bi-directional Recurrent Neural Network(RNN);
    for $Ts = 1$, $Ts{+}{+}$, *while* $Ts < Tz$ **do**
        Forward RNN: Compute hidden state $\vec{f} = (\vec{h_1}, \vec{h_2}, \vec{h_3}..\vec{h_{\tau_j}})$;
        // read the sentence in forward order
    for $Ts = z$, $Ts{-}{-}$, *while* $Tz < Ts$ **do**
        Backward RNN: Compute hidden state $(\overleftarrow{h}_1, \overleftarrow{h}_2..., \overleftarrow{h}_{\tau_j})$;
        // read the sentence in reverse order
    Compute annotation vector: $H_i = [\vec{h_j}^T; \overleftarrow{h_j}^T]$.;
for *Each input word* s_z *map through hidden states* $\vec{h_\tau}$ *and* $\overleftarrow{h}_\tau$ **do**
    Compute embedding lookup: $\vec{H_j} = f(\vec{h_{i-1}}, \bar{E}W_{s_n})$;
The encoder computations are deeply stacked in the following manner;
Compute for first layer **if** *i=1* **then**
    $h_{t,1} = f_1(h_{t-1}, 1, w_s t)$ // $h_{t-1,i}$:previous time stamp value
    else $h_{t,i} = f_{h_{t-1,i}, h_{t,i-1}}$;
    // $h_{t,i-1}$:previous layer in sequence value.
Compute context vector s_i *as summary of input sentence from Step 9 and 11*

Algorithm 2: Generation of next hidden state using Gated Recurrent Unit(GRU) Algorithm

input : Previous state h_{t-1} and input w_{s_t}
output: Next hidden state h_t

Forward States: Bi-directional Recurrent Neural Network;
// Gated Recurrent Unit(GRU) is designed in a manner to have more persistent memory thereby making it easier for RNN to capture a long-term dependency.
foreach *s(i)* **do**
  for *i=1,i¿z,i++* **do**
    $\vec{up}_i = \sigma(\vec{W}_{up}\bar{E}_{s_i} + \vec{O}_{up}\vec{h_{i-1}})$;
    // d=dimensionality of word embedding and u is number of hidden units $\bar{E} \in \mathbb{R}^{d \times k_s}$
    $\vec{res}_i = \sigma(\vec{W}_{res}\bar{E}s_i + \vec{O}_{res}\vec{h_{i-1}})$;
    // $\vec{W}, \vec{W}_{up}, \vec{W}_{res} \in \mathbb{R}^{u \times d}$
    $\vec{h}_i = tanh(\vec{W}\bar{E}s_i + \vec{O}[\vec{res}_i \odot \vec{h_{i-1}}]$;
    // $\vec{O}, \vec{O}_{up}, \vec{O}_{res} \in \mathbb{R}^{u \times u}$
    $h_i = (1 - \vec{up}_i) \odot \vec{h_{i-1}} + up_i \odot \vec{h}_i)$;
    // σ is a logistic sigmoid function.
  Backward states of bidirectional recurrent Neural network **for** *i=1,i¿z,i++* **do**
    $\overleftarrow{up}_i = \sigma(\overleftarrow{W}_{up}\bar{E}_{s_i} + \overleftarrow{O}_{up}\overleftarrow{h_{i-1}})$;
    $\overleftarrow{res}_i = \sigma(\overleftarrow{W}_{res}\bar{E}s_i + \overleftarrow{O}_{res}\overleftarrow{h_{i-1}})$;
    $\overleftarrow{h}_i = tanh(\overleftarrow{W}\bar{E}s_i + \overleftarrow{O}[\overleftarrow{res}_i \odot h - \overleftarrow{i} - 1]$;
    $h_i = (1 - \overleftarrow{up}_i) \odot \overleftarrow{h_{i-1}} + up_i \odot \overleftarrow{h}_i)$;
    $h_i = [\vec{h}_i + \overleftarrow{h}_i]$ // Combining forward and backward states

Algorithm 3: Embedding of linguistic features extracted from pipeline rule-based architecture algorithm

input : Each feature with distinct word embedding s_zy and
output: Context Vector s_i as summary of input sentence along with lingusitic features

Combining all these word vectors s_zy form a feature embedding matrix $E \in \mathbb{R}^{dy \times ky}$;
// dk is summation of dimension of all feature embedding and ky as vocabulary size of K^{th} feature
These embeddings are concatenated with total embedding size as the length matches. The input embedded sentence vectors are multiplied with these extracted linguistic features.;
// K_h is the vocabulary; and x is the output sentence length
for $s_zy = 1$, *s_zy++, while* $s_{zy} < z$ **do**
    $h_l = tanh(\vec{W} \prod_y^F \bar{E}_y s_{zy} + \vec{O}\vec{h}_{l-1})$;
    for *i=1,i¿zy,i++* **do**
        Calculates association between input word W_s to produce the next output wordw_h by calculating the impact of word representation($\vec{h_i}, \overleftarrow{h_i}$)
        $h_j = (\vec{h_i}, \overleftarrow{h_i})$;
for *i=1,i¿z,i++* **do**
    for *j=1,j¿z,j++* **do**
        Calculate alignment model a_{ij}// Output position around i to input position around j.
        Hidden state d_{i-1} and h_j // j^{th} annotation of input Sanskrit sentence.
        $a_{ij} = J_a^{\tau} tanh(W_a d_{i-1} + O_a h_j)$ // $W_a \in \mathbb{R}^{n_1'}, O_a \in \mathbb{R}^{n' \times n}, J_a \in \mathbb{R}_{n' \times 2n}$ are weight matrices
        $\alpha_{ij} = \frac{exp(a_{ij})}{\sum_{y=1}^{Ts} exp(a_{iy})}$ $S_i = \sum_{j=1}^{ts} \alpha_{ij} h_j$ // $S = feed - forwardNeuralNetwork$
        // The computed scalar attention value is normalized using softmax activation function, so all input words s adds up to 1.

Algorithm 4: Decoding the target sentence from context vector of linguistic features extracted from pipeline rule-based architecture

input : Previous hidden state d_{i-1} ,some representation of input context s_i and embedding of previous word output $E_{h_{i-1}}$
output: New output decoder hidden stated_i

for *i=1,i¿z,i++* **do**
  if *d=o* **then**
    $d_0 = f(W_d\vec{h_1})$
  // For intial hidden state $W_d \in \mathbb{R}^{d\times d}$.
  else $d_i = tanh(\bar{W}Eh_{y_{i-1}}) + O[res_i + d_{i-1}] + Ss_i)$;
  // hidden state d_i is computed given annotation from encoder and for update and Reset
  $up_i = \sigma(W_{up}E_{h_i-1} + O_{up}d_{i-1}S_{up}s_i)$;
  $res_i = \sigma(W_{res}Eh_{i-1} + O_{res}h_{i-1} + S_{res}S_i)$;
  // E= Embedded matrix of word for target language, u=Number of hidden units,d=word embedding dimension. $W, W_{up}, W_{res} \in \mathbb{R}^{u\times d}$
  // $0, O_{up}, O_{res} \in \mathbb{R}^{d\times 2d}$ are weight matrices.

Algorithm 5: Decoding the target sentence from context vector of linguistic features extracted from pipeline rule-based architecture

input : Decoder hidden state d_{i-1}, input context s_i and embedding of previous output word h_{i-1}
output: New word prediction w_{h_i}

The vector for prediction p_i for a output word;
for *i=1,i¿z,i++* **do**
  for *j=1,j¿l,j++* **do**
    $p_i = [maxp_i, 2\bar{j} - 1, i, \bar{2}j]^{\tau}_{j=1\ldots,l}$ // Repeat $E_{W_{h_i-1}}$ for d_{i-1} rather than d_i as it fragments the encoder state progress from d_{i-1} to d_i for prediction of output word p_i
    foreach *Output Word wh* **do**
      Calculate w_{h_i} as highest value in the vector $p_i = [maxp_i, 2\bar{j} - 1, i, \bar{2}j]^{\tau}_{j=1\ldots,l}$ // Training is performed accordingly as network being aware of correct output w_{h_i} assignes larger probability value
    $prob(h_i|d_{i-1}, s_i) \propto (h^{\tau}W_o pi)$ // Activation function softmax is used to convert raw vector into a probability distribution having sum of values as 1
    // ReLu combines input to yield the next hidden state

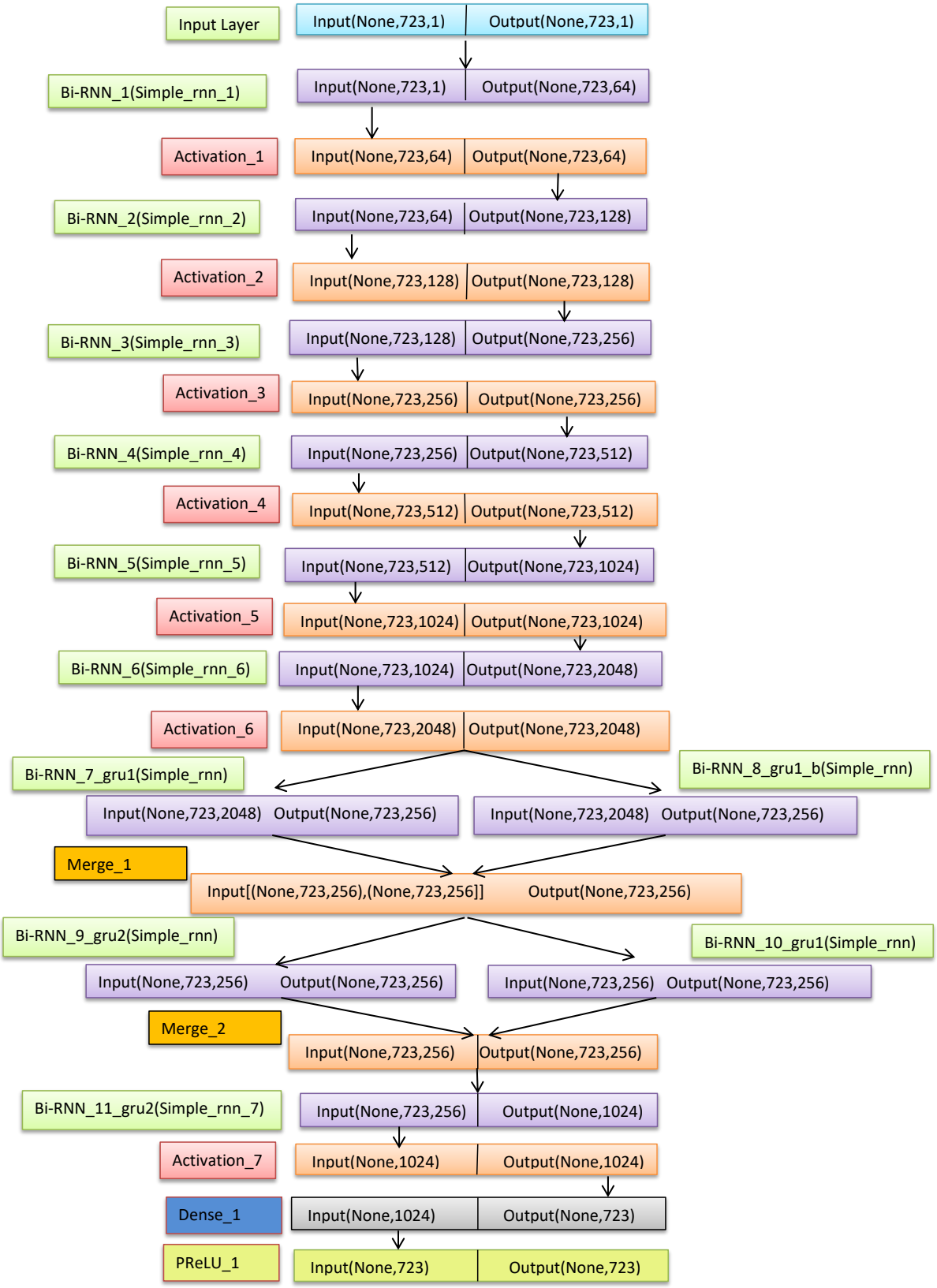


Figure 3.2: Flow Diagram of Neural Model

Table 3.1: Parallel and Monolingual Dataset from Different Domains

Domain	Parallel Corpus	Monolingual Corpus
News	25,000	202,269
Health care	NA	5,000
Tourism	NA	15,395
Literature	28,760	50,000
Wikipedia	NA	259,305
Judicial domain	NA	152,776
General Domain	49,000	36,000

Table 3.2: A Glimpe of Manually Curated Bhagavad-Geeta Parallel Corpus

	Sanskrit Sentences	**Hindi Sentences**
1	गाण्डीवं स्रंसतेहस्तात्त्वचैवपरिदह्यते। न च शक्नोम्यवस्थातुंभ्रमतीव च मेमनः।।	हाथ से गण्डीव धनुषगिररहाहैऔरत्वचाभीबहुतजररही ह ैतथामेरामनभ्रमित–साहोरहाहै इस लियेमैं खडारहनेकोसमर्थ नही हु।
2	निमित्तानि च पश्यामिविपरीतानिकेशव। न चश्रेयोऽनुपश्यामिहत्वास्वजनमाहव।।	हेकेशव! मैलक्षणों का भीविपरितही देख रहाहुँतथा युद्ध मेंस्वजनसमुदाय कोमारकरकल्याणभी नही देखता।
3	नकाङ्क्षेविजयंकृष्ण न च राज्यंसुखानिच। किंनोराज्येनगोविन्दकिंभोगैर्जीवितेनवा।।	हेकृष्ण! मै न तोविजय चाहताहुँऔर न राज्य सुखोंकोही।हेगोविन्द! हमें ऐसेराज्य से वयाप्रयोजनहैअथवा ऐसेभोगो से और जीवन से भीवयालाभहै।
4	येशामर्थेकाङ्क्षितंनोराज्यंभोगाः सुखानिच। तइमेऽवस्थिता युद्धे प्राणांस्त्यवत्या धनानिच।।	हमेंजिनकेलियेराज्य,भोगऔरसुखादिअभीष्टहैं, वेही ये सब धनऔर जीवन की आषाकोत्यागकर युद्ध मे खडेहैं।
5	आचार्याः पितरः पुत्रास्तथैव च पितामहाः। मातुलाः श्वषुराः पौत्राः श्यालाः सम्बन्धिनस्तथा।।	गुरूजनताउ–चाचे, लडकेऔरउसीप्रकारदादे, मामे, ससुर, पोत्र सालेतथाऔरभी सम्बन्धीलोगहै।
6	एतान्नहन्तुमिच्छामि घ्नतोऽपि मधुसूदन। अपि त्रैलोक्यराज्यस्य हेतोः किनुमहीकृते।।	हे मधुसूदन !मुझे मानेपरभीअथवातीनोंलोकों के राज्य के लियेभीमैंइनकोमारनानहीचाहता! हफारपृथ्वी के लियेतोकहनाहीक्याहै।
7	निहत्य धार्तराष्ट्रान्नः काप्रीतिः स्याज्जनार्दन। पापमेवाश्रयेदस्मान्हत्वैतानाततायिनः।।	हेजनार्दन ! धृतराश्टपुत्रोंकोमारकरहमेंक्याप्रसन्नताहोगी ! इनआततायियोंकोमारकरतोहमेंपापहीलगेगा।
8	तस्मान्नार्हावयंहन्तुं धार्तराष्ट्रान्स्वबान्धवान्। स्वजनंहिकथंहत्वासुखिनः स्याममाधव।।	अत एवहेमाधव ! अपनेहीबान्धव धृतराश्ट्र के पुत्रोंकोमारने के लियेहम योग्य नही है:, क्योंकिअपनेहीकुटुम्बकोमारकरहमकैसेसुखीहोगें।
9	यद्यप्येते न पश्यन्तिलोभोपहतचेतसः। कुलक्षयकृतंदोशंमित्रद्रोहे च पातकम्।। कथं न ज्ञेयमस्माभिः पापादस्मान्निवर्तितुम्। कुलक्षयकृतंदोशंप्रपश्यद्भिर्जनार्दन।।	यद्यपि लोभ से भ्रश्टचितहुए ये लोगकुल के नाष से उत्पन्नदोशकोऔरमित्रों के विरोध करनेमेंपापको नही देखते, तोभीहेजनार्दन! कुल के नाष के उत्पन्नदोशकोजाननेवालेहमलोगोंइसपाप से हटने के लियेक्यो नही विचारकरनाचाहिये।
10	कुलक्षये प्रणश्यन्तिकुलधर्माः सनातनाः। धर्मेनश्टेकुलंकृत्स्नमधर्मोऽभिभवत्युत।।	कुल के नाष से सनातनकुल–धर्मनश्टहोजातेहैं, धर्म के नाषहोजानेपरसम्पूर्णकुल मे पापभीबहुतफैलजाताहै।
11	अधर्माभिभवात्कृश्णप्रदुश्यन्तिकुलस्त्रियः। स्त्रीशुदुश्टासु वार्श्णेय जायतेवर्णसंकरः।।	हेकृश्ण! पाप के अधिक बढ जाने से कुल की स्त्रियांअत्यन्तदूशितहोजातीहैऔरवार्श्णेय! स्त्रियों के दूशितहोजानेपरवर्णसंकरउत्पन्नहोताहै।
12	संङ्करोनरकायैवकुलघ्नानांकुलस्य च। पतन्तिपितरोह्येशांलुप्तपिण्डोदक्रियाः ।।	वर्णसंकरकुलघातियोंकोकुलकोनरकलेजाने के लियेहोताहै । लुप्तहुईपिण्डऔर जल की क्रियावालेअर्थात् श्राद्ध औरतप्रणयचितइसकेपितरलोगभी अधोगतिकोप्राप्तहोतेहैं।
13	दोशैरेतैः कुलघ्नानांवर्णसंङ्करकारकैः। उत्साद्यन्तेजातिधर्माः कुलधर्माश्च शाश्वताः।।	इसवर्णसंकरकारकदोशों से कुलघातियों के सनातनकुल–धर्मऔरजाति–धर्मनश्ट होजातेहैं।
14	उत्सन्नकुलधर्माणांमनुश्याणांजनार्दन। नरकेऽनियतंवासोभवतीत्यनुशुश्रुम।।	हेजनारर्दन !जिनकाकुल धर्मनश्टहोगयाहै, ऐसेमनुश्योंकाअनिच्छिमकालतक नारीमेंवासहोताहै, ऐसाहमसुनतेआयेहैं।
15	अहोबतमहत्पापंकर्तुव्यवसितावयम्। यद्राज्यसुखलोभेनहन्तुंस्वजनमुद्यताः।।	हां! षोक! हमलोगबुद्धिमानहोकरभीमहान् पापकरनेकोतैयारहोगयेहैं, जो राज्य औरसुख के लाभ से स्वजनोंकोमारने के लियेउद्यतहोगयेहै।
16	यदिमामप्रतीकारमशस्त्रं शस्त्रपाणयः। धार्तराश्ट्रारणेहन्युस्तन्मे क्षेमतरंभवेत्।।	यदिमुझ षस्त्ररहित एवसामना न करनेवालेकोहाथमेंलियेहुए धृतराश्ट्र के पुत्र रणमेंमारडालेतोवहमारनाभीमेरेलियेअधिकवलयाणकारकहोगा।
17	एवमुक्त्वार्जुनः संङ्ख्ये रथोपस्थउपाविशत्। विसृज्य सशरंचापं शोकसंविग्नमानसः।।	संजय बोले–रणभूमिमें शोक के उद्विग्न मनवालेअर्जुनइसप्रकारकहकरबाण सहित धनुशकोत्यागरकरथ के पिछलेभागमेंबैठगये।

Table 3.3: Additional Monolingual Dataset

# Lines	221528
#Words	2849514
#Characters	38413350
#Total	2.8 Million

Table 3.4: Dataset Division into Training, Development and Testing

Dataset	Sentences	Words		Vocabulary	
		Source	Target	Source	Target
Training	145,34,215	131575835	123425654	355.465	124.278
Development	192679	172799	122645	NA	NA
Test	12698	77322	26273	NA	NA

3.2.2 Model Size

The Neural model along with the extracted linguistic features trained on tensorflow platform consists of various parameters. The dimensionality of each parameter is reflected in Table 3.5.

Table 3.5: Model Size

Parameter	Dimensionality
Encoder(forward and backward unit each)	1000
Decoder	1000
x (Word Embedding)	1000
n (GRU hidden state)	1000
d(dimension of word embedding)	620
v(Output maxout hidden layer)	500
n‘(alignment model hidden units)	1000
W_s	32×400
W_h	32×1
W_{up}	64×400
W_{res}	65×2

3.2.3 Parameter Initialization

The parameters used to train the model are exhibited in Table 3.5. The weight matrices have been used recurrently as random orthogonal matrices. The bias component has been omitted for forming simpler equations. All, the alignment elements (V_a and bias component) were initialized as 0. The alignment matrices were initialized with $variance = 0.001$ and $mean = 0$ from Gaussian distribution. All other matrices were initialized with the same mean with a variation in $variance = 0.01$.

3.2.4 Training

In the proposed work, Keras sequential model has been used to process the data. The proposed model is processed through a highly configured core GPU with 32 GB of RAM to achieve a high throughput speed of approximately 2500 words per second. This speed is not possible for normal systems because in this one epoch takes approximately two hours to run. Thus, a highly configured GPU along with NVIDIA Geforce GTX 1050 and Quadro K6000 was used for the purpose. Each epoch passed over the training set and test set as shown in Figure 3.3, and updates were performed for each minibatch parameters. The training and development probability was the average conditional log-probability of the sentence to be in either of the sets.

Vanilla Stochastic Gradient Descent(SGD) algorithm was used with automatically updating the learning rate using Adadelta[241](parameters$\rho = 0.95$ and $\epsilon = 10^{-6}$). Adam

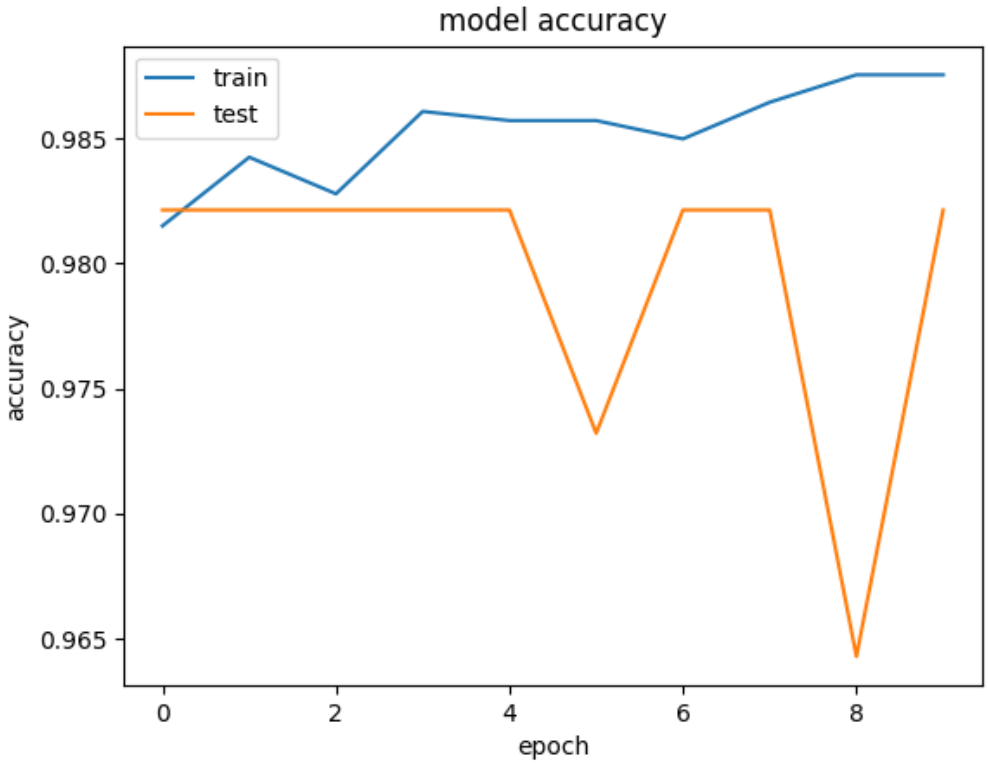


Figure 3.3: Epochs for Training and Test Set

optimizer was employed for stochastic optimization[242]. Normalization is performed[243] for each of the mini-batch(distributed data set).As the distribution of input layer changes due to the change in the parameter of the previous layer during the training, it makes training difficult to perform. Normalization is conducted to reduce the internal covariate shift and it increases the learning rate by reducing the initialization process. It even reduces the need for dropout by acting as a regulariser. A random size for minibatch of 64 sentences was taken which normalized after exceeded the threshold value of 1. Each update took time equivalent to its longest sentence. To minimize the time, sentences were sorted and shuffled manually by retrieving 1500 pair sentences after every 20^{th} update.

3.3 Performance Evaluation

The performance of the proposed SHH-MTS has been evaluated using automated metrics and human evaluators. The sentence length of the corpora is affecting the updating of the model. It is evident from Figure 3.4(a) that on increase in the sentence length in the training corpus, the number of updates of weights in the model training drastically increases over a point (till 20 words length) and then decreases after the sentence length exceeds this limit. Figure 3.4(b) highlights the effect of sentence length on several iterations performed on the training set, i.e. epochs. It can be deduced from the graph that

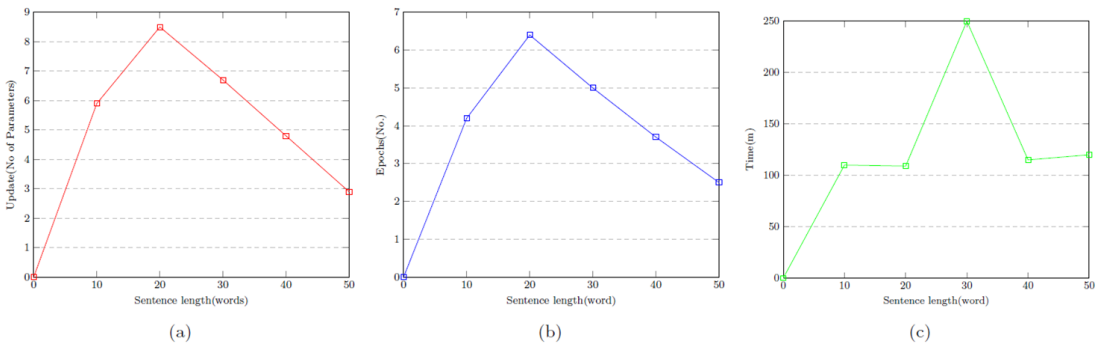


Figure 3.4: Sentence Length Affecting (a) Updates,(b) Epochs, and (c) Time

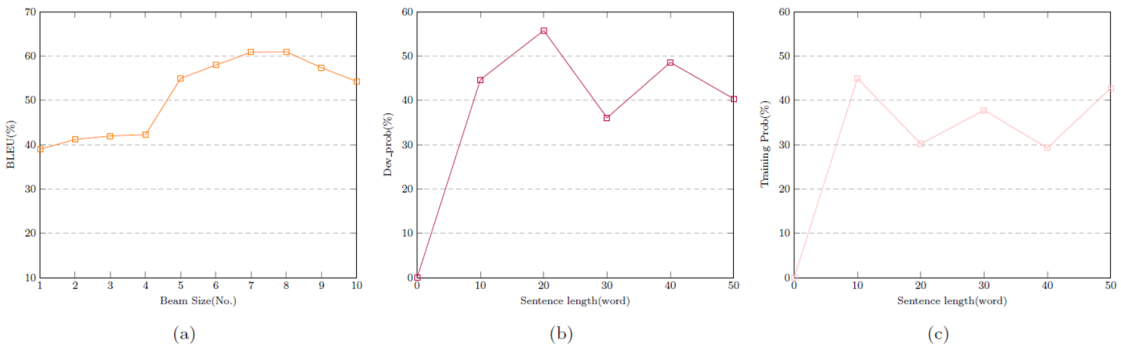


Figure 3.5: (a) BLEU Varies with Beam Sizes (b) Development Probability Varies Depending on Sentence Length (c) Training Probability Varies Depending on Sentence Length.

the number of epochs decrease's after the point (i.e., 20 sentence length). Figure 3.4(c) denotes the effect of sentence length on time for building the model. It is clear from the figure that, the time fluctuates drastically. For 10-20 sentence length, the model training time remains the same, whereas for 20-30 sentence length it increases rapidly. Finally, the sentence length up to 20 words limit the update, epochs and time. If the sentence length in corpus exceeds this limit, a drop can be seen as shown in the plotted graphs. Figure 3.5(a) depicts that the BLEU score varies with the beam size. A beam search was used during the inference to find the most likely sequence of words for each translation. The beam problem in Neural Machine Translation exists for relatively small beam sizes – especially when compared to traditional beam sizes in Statistical Machine Translation systems. The figure shows that beam size (1-4) have a constant change in the BLEU score, whereas from 5-10 the BLEU score is enhanced. If there is a large increase in the beam size, it drops the BLEU score. So, here in our experiment training, limited beam size by normalizing the length of sentences. Figure 3.5(b) shows the effect of sentence length on the development probability. As shown in the plot, the development probability increases only up to 20 sentence length, and thereafter it decreases. Therefore, it would be ideal either to split the longer sentences or normalizing the sentences of length more than 20 lengths. Figure 3.5(c) models the sentence length effect on the training probability. As already explained, the training probability decreases with time. So, to increase the

training probability shorter sentences would help. The different automated metrics such as BLEU, Word Error Rate(WER), F-measure and Meteor used in the current work have been discussed in the following section.

3.3.1 Automatic Error Analysis

- BiLingual Evaluation Understudy(BLEU) is an important metric used for calculating the accuracy of translated sentences as compared to the human-generated reference translations as in Eq.(5.1). It provides accurate results for longer sentences but fails in the case of shorter sentences[244]. The BLEU score is evaluated at each iteration of performance enhancement as exhibited in Table 3.6. Firstly, a simple sequential model was built which gave an accuracy of 10.23%. To improve the accuracy of the model, Keras model with a bidirectional layer was applied which significantly improved the accuracy to 29.12%. The result produced a readable translation, but it required further improvement. It was achieved through Gated Recurrent Unit(GRU) cells which performed better computation. To attain the output, the activation function was applied in the neural model, but with the implementation of different activation functions, the accuracy of achieved i.e., 56.78% was achieved for the proposed system. The accuracy came to a stagnant level with all the significant experimentation, but auto-tuning slightly improved. The accuracy of the proposed model which finally came to 61.02%. It will compute precision w.r.t. human-generated translation without taking into account any grammatical corrections/errors.

$$BLEU = min(1, \frac{output\_length}{Reference\_length})(\prod_{i=1}^{4} precision_i) \tag{3.28}$$

As experimented with different models to enhance the performance of MTS.The BLEU

Table 3.6: BLEU Scores of Different Experiments Performed

Models	**BLEU(%)**
Simple Sequential Model	10.23
Keras Model with bi-directional layers	29.12
Keras Model + bi-directional layer + gru	40.34
Keras Model + bi-directional layer + gru+ Relu and sigmoid activation function	56.78
Keras Model + bi-directional layer + gru+ Relu and sigmoid activation function + auto-tuning	61.02

score computed for all the three models, i.e. RBMT, Neural and Hybrid depicted in Figure 3.6(a). The proposed model was built in three phases; firstly, the rule-based model; secondly neural model; and finally, hybrid model combining other models which

performed better than rest of the models. The results demonstrate the efficiency of this novel technique applied in the proposed work. The BLEU score obtained for the proposed system varied with the beam size as displayed in Figure 3.5(a). The development probability varied with the sentence length as shown in Figure 3.5(b); and the training probability depended on sentence length.

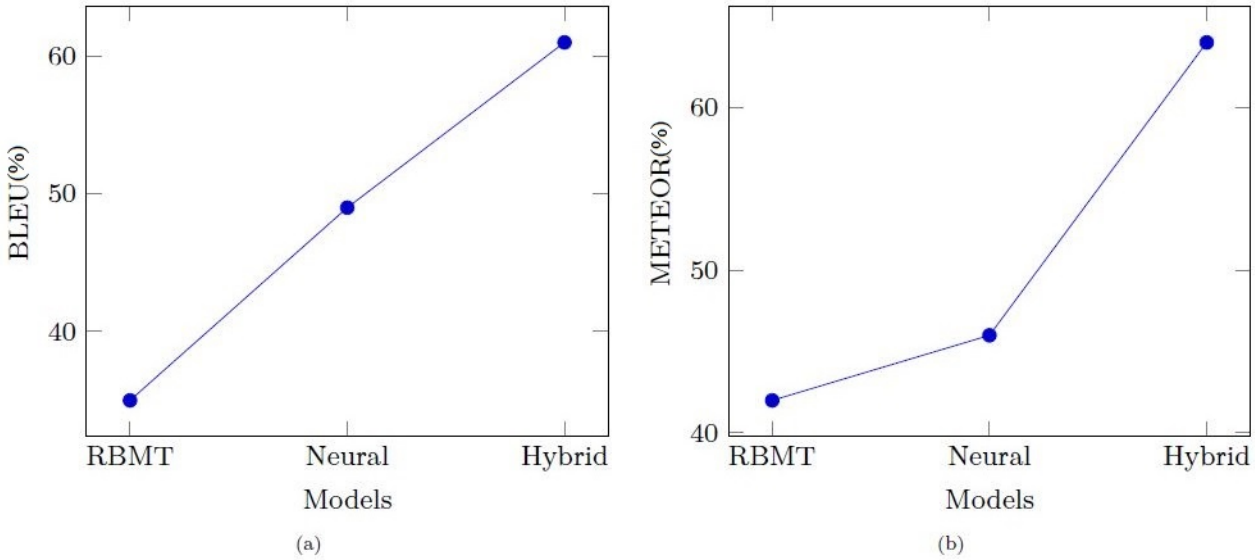


Figure 3.6: RBMT, Neural and Hybrid Models Across their (a) BLEU (b) METEOR

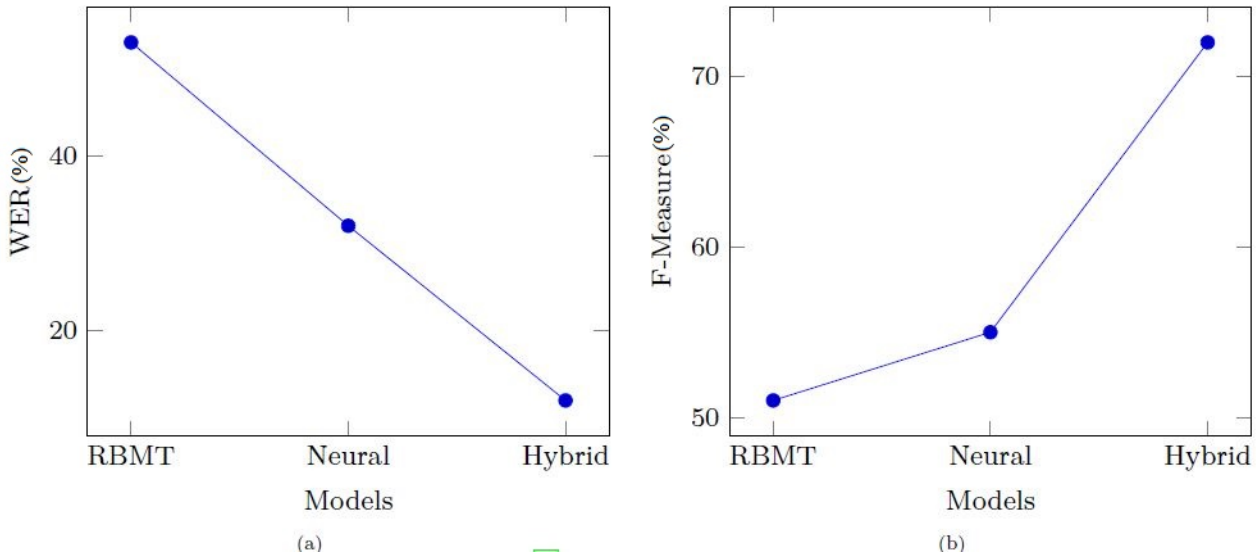


Figure 3.7: RBMT, Neural and Hybrid Models Across their (*a*) Word Error Rate(WER) (*b*) F_measure

- Word Error Rate (WER): It is a metric used to calculate the error rate by comparing MT output with the human translated output as in Eq.(5.2). The lower is the WER,

better would be the model.

$$WER = \frac{substitutions + insertions + deletions}{reference\_length} \tag{3.29}$$

Here, substitution means replacement of one word with another in a particular sentence. Insertion means the addition of words; and deletion signifies the dropping of words. The WER score computed for all the three models, i.e. RBMT, Neural and Hybrid is reflected in Figure 3.6(b).

As already explained the model was built in three phases. Firstly, the rule-based model was built as already explained in Chapter 3; secondly, neural model was prepared; and finally, hybrid model was built by combining other models which performed better than rest of the models. The results proved the efficiency of this novel technique applied in the proposed work. It can be hypothesised from the figure that hybrid model has a minimum WER score. Further, BLEU and WER are inversely proportional to each other.

- F-measure: It is a metric used for calculating the accuracy and precision of the model as in Eqs.(5.3-5.5). It calculates the quality or exactness of an output. Mathematically, the calculation of F-measure requires precision and also the recall values. Thus,

$$Precision = \frac{Correct}{Output\_Length} \tag{3.30}$$

$$Recall = \frac{Correct}{reference\_length} \tag{3.31}$$

F-measure

$$F - measure = \frac{(Precisionrecall)(Precision + Recall)}{2} \tag{3.32}$$

The F-measure computed for all the three models, i.e. RBMT, Neural and Hybrid is depicted in Figure 3.7(a). The proposed model was built in three phases; the rule-based model as explained in Chapter 3, neural model as described in Chapter 3 and hybrid model prepared by combining other models which performed better than rest of the models. The results have demonstrated the efficiency of this novel technique applied in the proposed work. It has been observed that the hybrid model has greater F-measure as compared to other existing models.

- METEOR: It is used to find the correlation between the machines translated output and the human-generated sample output as in Eqs.(5.6-5.8). This score is also directly proportional to accuracy.

Reducing the effect of F-mean is helpful[245].

$$F - mean = \frac{10PR}{9 + RP} \tag{3.33}$$

Here P stands for Precision, and R for Recall. F-mean, Precision and Recall are based on the unigrams matches. For longer values, penalty requires computation. Mathematically,

$$Penalty = 0.5(\frac{chunks}{unigrams\_matched})^3 \tag{3.34}$$

$$Score = F - mean \cdot (1 - penalty) \tag{3.35}$$

The METEOR scores computed for all the three models, i.e. RBMT, Neural and Hybrid are presented at a glance in Figure 3.7(b). The result demonstrates the efficiency of our novel technique applied in our proposed work. In the proposed hybrid model, meteor value is higher as compared to other models due to a strong correlation between the words of the output sentences. It can be concluded that, the hybrid model has refelcted higher BLEU, F-score and Meteor, but the WER value has been lesser as compared to other models.

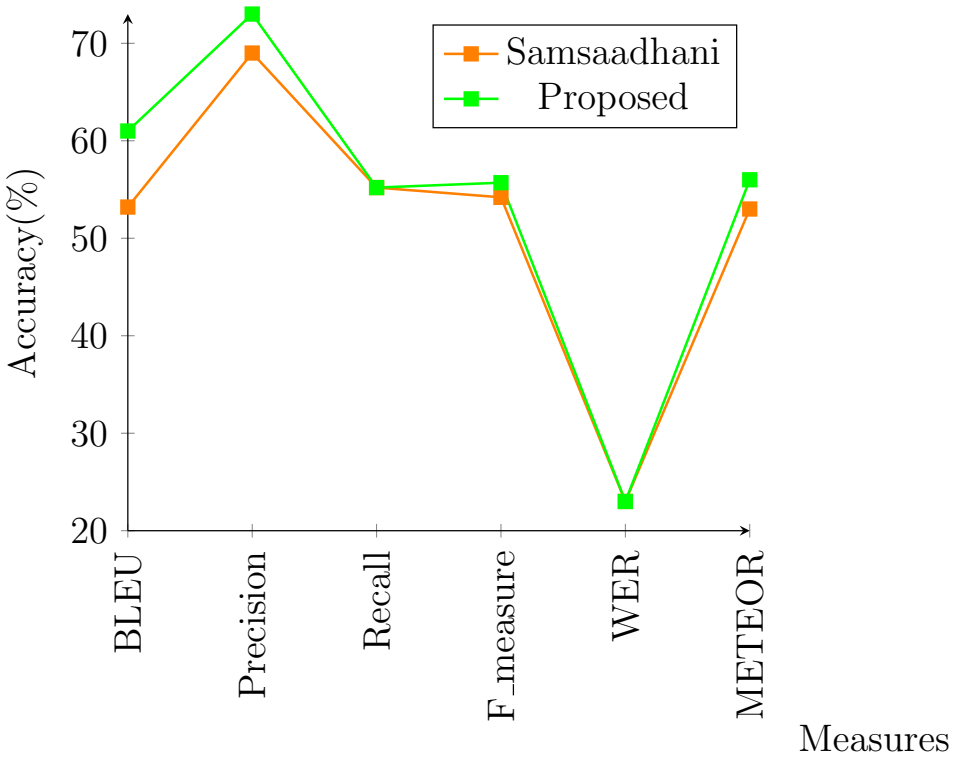


Figure 3.8: Comparison of Baseline System, i.e. RBMT for Sanskrit-Hindi[212] Corresponding to the Proposed System on Various Evaluation Measures

The performance analysis was also conducted on different categories of Sanskrit sentences consifering the various automated metrics such as BLEU, Precision, Recall, F-measure, WER, F-mean, Penalty and Meteor as shown in Table 3.7. These sentences were run on the developed system provides with Hindi output sentences used for performance evaluation along with their reference translation.

3.3.2 Human Error Analysis

In this work, linguistic errors were identified by performing a case study. It included 15 different grammatical cases corresponding to which Sanskrit sentences are tested. The output generated by passing different kinds of sentences is presented in Tables 3.8 and 3.9 . These results showed that the highest error rate in the proposed hybrid MTS was encountered in the sentences of category verb 4% whereas other categories have less than 3% of error rate.

3.3.3 Comparison of the Proposed System with Existing Work

The proposed work makes a comparison with the existing work [246][190] and [247]. The comparison undertaken on the basis of automated metrics is depicted in Figure 3.8 and on human evaluation in Figure 3.9. The comparative analysis is presented in Figure 3.10(a). It is evident from Figure 3.10(b) that the error rate of the Sanskrit-Hindi Statistical Machine Translation (SaHit)[190] is higher as compared to the proposed system. It was also compared with the recent research work carried out by [247] as in Figure 3.10(c). The proposed hybrid MTS has recorded an accuracy of 61% which is higher than that of the existing systems developed[190] and [247] with the respective percentages as 27% and 57%. The result were also compared through human evaluation based on grammatical categories, i.e. Sandhi, Compound, Verb, Over Generation, Less Generation, Visarga/Anusarva, Karaka and others. The proposed system was more readable, and provided better output than the existing work for this domain. The proposed work has also been compared on the basis of its adequacy and fluency with the existing work carried out by [247] as shown in Table 3.10. The proposed work for Sanskrit language processing is a novel technique achieving greater accuracy in terms of both automated and human analysis.

3.4 Conclusion

In this chapter, three different translation models(Rule-based, Neural-based and hybrid) produced have analysed for Sanskrit to Hindi. In comparison to the earlier works [[190][247][248] which lack extensibility, generalizability and adaptability, the proposed

Table 3.7: Metric Analysis of Sanskrit to Hindi Translation

Sanskrit	Hindi	Reference Translation	BLEU	Precision	Recall	F-measure	WER	F-mean	Penalty	Meteor
साध्वी: शीघ्रं मैत्री भवति	साधु जल्दी से दोस्त बन जाते हैं	अच्छे लोग जल्दी से दोस्त बन जाते हैं	75%	75%	75%	75%	50%	75%	18.50%	61%
हरे: पत्नी लक्ष्मी:	हरि की पत्नी लक्ष्मी है	लक्ष्मी हरि की पत्नी है	77%	77%	77%	77%	30%	77%	18.40%	63%
आपत्काले बुद्धे: परीक्षा भवति ।	आपत्काल में बुद्धि की परीक्षा होती है	आपातकालीन में बुद्धि का परीक्षण होता है	70%	70%	70%	70%	50%	71%	18%	59%
तत्र धेनूनां समूह: तिष्ठति	वहाँ गायों समूह रहता है	गायों का समूह वहां रहता है	84%	82%	80%	79%	10%	79%	17%	79%
स्यात् स्वप्न: आकाशात् उच्चतर: सागरात् गभीरतरश्च ।	आकाश की तुलना में अधिक ऊँचा ड्रीम और महासागर से गहराई से	ख्वाब आकाश से ऊँचा और महासागर से गहरा होना चाहिए	69%	64%	63%	66.78%	60%	62%	19%	52%
अहम् गच्छामि	मैं जाता हूँ	मैं जाता हूँ	99%	99%	99%	98.78%	100%	100%	99.90%	98.70%
यदा अशान्तिवृत्ति: तदा शान्तिसमुद्भव: ।	शांति तब शुरू होती है जब अपेक्षा समाप्त होती है	जब अपेक्षा समाप्त होती है तब शांति शुरू होती है	87%	85.60%	80%	81.20%	13.40%	79%	27%	81%
वचांसि हन्तुं प्रभवन्ति । वचनप्रयोगे भव अप्रमत्त: ।	शब्द मार सकते हैं सावधान रहें जब आप उन्हें प्रयोग कर रहे हों।	शब्द मार सकते हैं उनका प्रयोग सावधानी से करें ।	61%	62%	61.20%	59%	54%	69%	84%	53%

Table 3.8: A Case Study of Linguistic Analysis of Sanskrit to Hindi Translation-I

Case No.	Type	Input	Output
Case 1	Anusvara and Visarga Sentences	ब्राह्मणाय भूलोके सर्वम् अनित्यम् देवाःधनं ददाति	ब्राह्मण के लिये भूलोक में सब को अनित्य को देवाःधनम् देता है
Case 2	Noun	भवन्तः सर्वे एतं मार्गं अनुसृत्य धर्मं, राज्यं, प्रजाः च रक्षन्तु इति	आप सब इसको मार्ग को अनुसरण करके धर्म को राज्य स्वामी और रक्षा करिए ऐसा
Case 3	Sandhi Sentence	नृपो□स्य नगरस्य बहिर्वनं गत्वा पुनस्तस्मात् वनान्नैत्यत्यागच्छत्	नृपो□स्य नगर का बहिर्वनम् जा कर पुनस्तस्मात् वनान्नैत्यत्यागच्छत्
Case 4	Locative Absolute	सैनिकेषु युद्धेषु हतेषु	सैनिकों में युद्धों में मारा गयों में
Case 5	Compound Sentences	पूर्वस्मिन् काले कृतानां पापानां विपाकेन	पूर्व में समय में निर्मितों के पापों के खाना पकाने से
Case 6	Pronouns	कतिपयान् दिवसान् श्रुत्वा	कुछों को दिनों को श्रुत्वा करके
Case 7	Complex Sentences	विकसितानि चित्रवर्णानि पुष्पाणि परितः जनाः जनानां चित्तम् आह्लादितं कुर्वन्ति ।	विकसित चित्रवर्णानि पुष्प चारों ओर जन जनों का चित्त को आह्लादितम् करते हैं
Case 8	Infinitive Sentences	नृपस्य वचनं श्रुत्वा, सूतः यथा आज्ञापयति देवः इति उक्त्वा, शुभान् श्वेतान् अश्वान् रथे अयोजयत् ।	राजा का वचन सुन कर सूत जैसे आज्ञापयति देव ऐसा बोल कर शुभ को सफेद को अश्वों को रथ में चित्त स्थिर किया
Case 9	Gerund PPP Phrases	अन्धेन धूर्तेन तं अधिगम्य तद् एव उक्तम्	अन्य धूर्त के द्वारा उसको प्राप्त हो कर वह ही कहा हुवा
Case 10	Numeral sentences	सांख्य-दर्शनस अनुसारेण पञ्च-विंशतिः तत्तानां□वदनेयथा पुरुषः प्रकृ□तिः बुद्धिः अहंकारः मनः पञ्च जान-इन्द्रियाण पञ्च कर्मर-इन्द्रियाण पञ्च तनात्रीण पञ्च महाभूतीन च । बुद्धिः अहंकारः मनः दश इन्द्रियाण च त्रयो-दश अन्:करणाइन इति उच्यते ।	1.1 साङ्ख्यदर्शनस अनुसार से पञ्चविंशतिः तत्तानिवदनेयथा पुरुषः प्रकृ यहाँ से बुद्धिः अहंकार मन पञ्च जानइन्नयाइण पञ्च विषयीइन्नयाइण पञ्च तनाताइण पञ्च महाभूताइन और 2.1 बुद्धिः अहंकार मन दश इन्नयाइण और त्रयोदश अन:करणाइन इति उच्यते

Table 3.9: A Case Study of Linguistic Analysis of Sanskrit to Hindi Translation-II

Case No.	Type	Input	Output
Case 11	Anvaya sentences	शीभगवान्उवाच । पाथर, मेनानिवथिान ि◌द-विान नाना-वणर-आकृ तीिन च रपािण शतश: अथ सहसश: पश ।	1.1 शीभगवानुवाच 2.1 पाथर मेनानाइवथाइन इदवाइन नानावणरआकृतीइन और रपाइण सौ तरह से और सहसश: पश
Case 12	Comparative and Superlative Adjectives	नलंतका एव तेहंसा: वनात्समुतत ि◌वदभर-नगरीम्आगम दमयना: समीपे नपेतु: ।यदा सखी-गण-वृता दमयनी तान्सवेिषांपिकणांशेिषान्कनक-अलंकृ तान्हंसान्अपशात्तदा संतुष- मानसा एकं हंसंगहीतुंशेिथम्उपचक्रमे । अननरंतेसवेसवरत वनेिवससपु: एके कश: तुता: कना: तान्समुपादवन् । तत: यंहंससा दमयनी उपाथावत्स: मानषी वाचंकृ ता तामअबवीत् ।	1.1 नलन्तका ही तेहंसा: वनात्समुतत इवदभरनगरीमागम दमयना: समीपेइनपेतु: 2.1 जब पसन्द की दमयनी तान्सवेिषाम्पिकणांशेिषान्कनकअलङ्कृ तान्हंसानपशात्तदा मानसा एक को हंसङ्गाहीतुंशेिथमुपचक्रम 3.1 अननरन्तेसवेसवरत वनेइवससपु: 4.1 एके कोइ तुता: कना: तान्समुपादवन् 5.1 वहाँ से यंहंससा दमयनी उपाथावत्स: मानवी वाचङ्कृ ता तामबवीत्
Case 13	Causative Verbs and Denominative Verbs	पुरष-ऋषभ, एतिह यंसम-सुख-द:खं ◌, धीर, पुरषंन वथयिन, स: अमृतताय कलते ।	पुरषऋषभ एतेइह यंसमसुखद:खम् ऊ धीर को पुरषंन वथयिन वह अमृतताय कलतायें
Case 14	Meter Passive Stems and Gerundives Passive Impersonal	आचाय: ि◌पतर: पुता: तथा एव च ि◌पतामहा: मातुला: शशुरा: पौता: सालाः तथा संबिनन: ।थर्मक्षेत्रे कुरुक्षेत्रे समवेता युयुत्सव: ।मामका: पाण्डवाश्चैव किमकुर्वत सञ्जय ।।तदा इदंवाकंहषीके शंजाह महीपते । "अचुत सेनयो: उभयो: मेरथंसापय ।	1.1 अआचाय: इपतर: पुता: तथा एव च िपतामहा: मामा शशुरा: पौता: साल वैसे सम्बिनन: 2.1 थर्मक्षेत्र कुरुक्षेत्र एकत्रित लड़ने की इच्छा वाले 3.1 मेरे पाण्डवाश्चैव किमकुर्वत संजय तब इदंवाकंहषीके शंजआह महीपतायें 4.1 अचुत सेनाओं का उभका मेरथंसापय
Case 15	Conjugation sentences	तासांनदीनांसपमीगंगासयंस-सननम्आसतंभगीरथम्अनगचत् ।	1.1 तासांनदीनांसपमीङ्गासयंससननमआइसतम्भगीरथमनगचत्

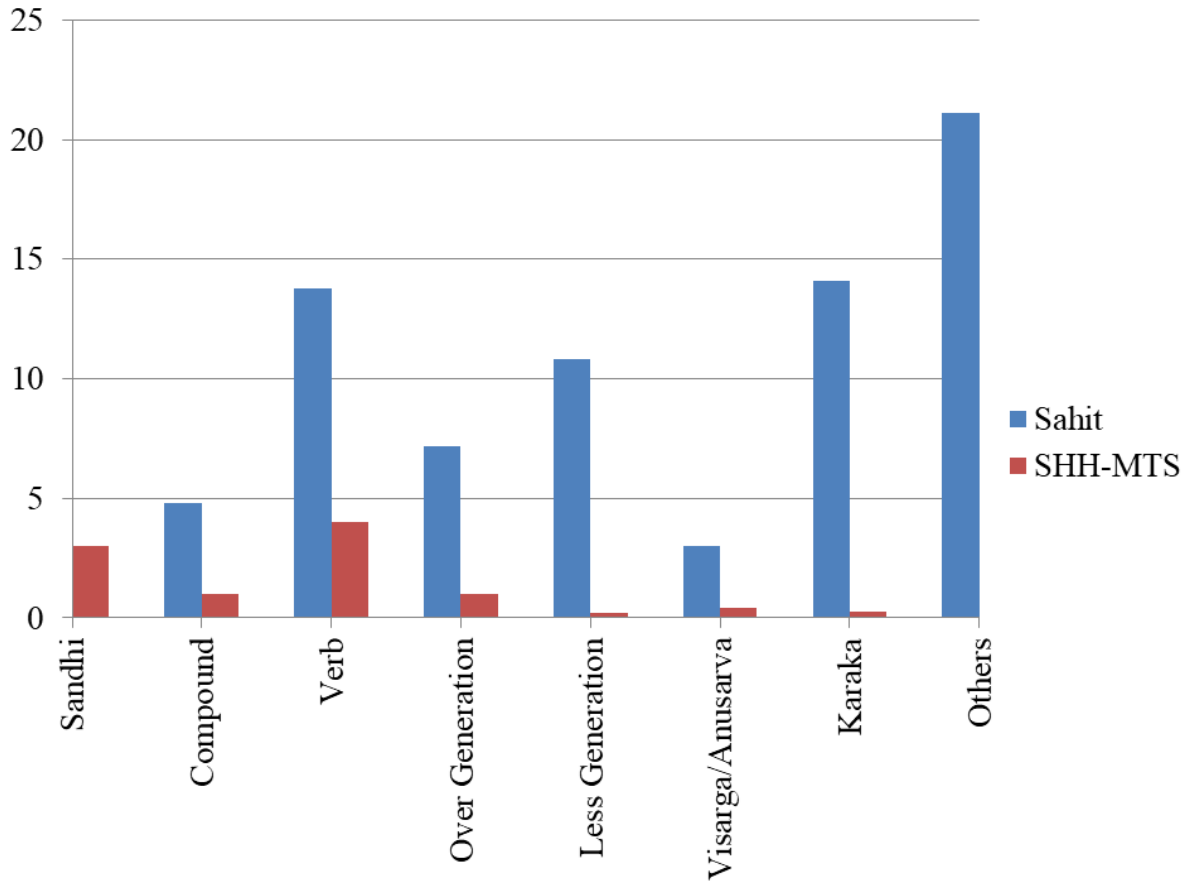


Figure 3.9: Human Evaluation of the Proposed System with Sahit

Measure	[247]	Proposed System
Adequacy	91%	96%
Fluency	66.72%	71.26%

Table 3.10: Adequacy and Fluency of the Proposed System and the Existing Work[247]

system is capable of overcoming these problems. It extracts features from the linguistic rule and further passes on these features to train the recurrent neural network. Performance evaluation performed on automatic and human measures has resulted in the excellent performance of the hybrid system. It has outperformed even in terms of accuracy, speed and response time. The proposed hybrid model is fast and more efficient than the existing rule-based systems. In non-rule match cases, the rule-based model does not return any output. While the proposed model always provides the best solution. The previous models become inefficient in the case of long sentences and are practically infeasible sometimes, but the proposed model is efficient in such scenarios. The work developed and presented here is novel and can be applied to any low-resource language pair with minimum linguistic knowledge. In future research, multiple linguistic languages could be considered to convert into the single target language and multi-lingual platform for the same purpose.

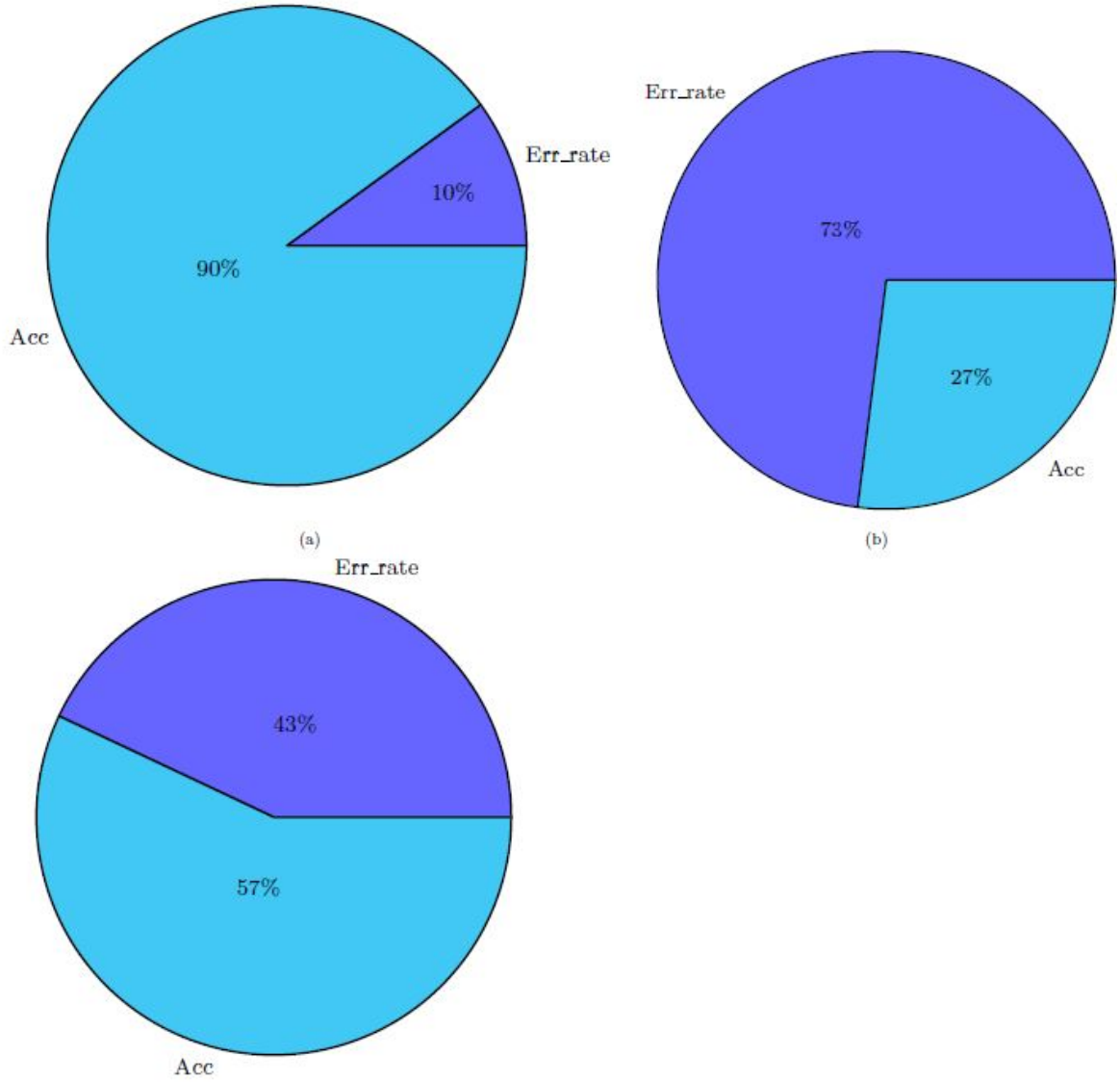


Figure 3.10: Comparison of Overall Error Rate and Accuracy (*a*) SHH-MTS and Existing Systems (*b*) [190], and (*c*) [247]

The next chapter presents the deployment of the developed stand-alone system in this chapter. It is deployed on the cloud to offer translation as a service.

Chapter 4

Deployment of Proposed Sanskrit-Hindi Machine Translation System on Cloud

The previous chapter presents the proposed system that overcame the earlier works for translating Sanskrit-Hindi lacks extensibility, generalizability and adaptability. The proposed system extracted features from the rule-based system as linguistic rules and feeds them further to train the recurrent neural network.

The current chapter presents the description for the deployment of the proposed work on the cloud. It offers translation as a cloud service by improving the quality of service (QoS) from the stand-alone system. The objective of this work lies in demonstrating the management of recurrent changes in terms of corpus, domain, algorithm and rules.

The chapter describes SHH-MTS as a service. Section 4.2 presents the deployment infrastructure for the proposed work while experimental details covered in Section 4.3. The results discussed in Section 4.4. At last, the chapter concludes in Section 4.5.

4.1 Sanskrit-Hindi Hybrid Machine Translation System(SHH-MTS) as a Service

This section presents the SHH-MTS deployed on cloud to offer as a service. As MTS improves performance through recurrent changes in terms of corpus, domain, algorithm and rules. These rapid changes are very difficult to be updated in stand-alone MTS. To facilitate this problem, it is proposed that the system must be deployed on cloud to provide Quality of Service(QoS) to the end-user. First of all, Sanskrit-Hindi Hybrid Machine Translation System(SHH-MTS) architecture was designed and developed. Later on, after its development, it was deployed on cloud to test and analyze its performance on various parameters such as throughput, server load, response time, and CPU utilization.

4.1.1 Characteristics of Sanskrit-Hindi Machine Translation System

These are summarized as follows:

- The hybrid machine translation system developed for converting Sanskrit to Hindi languages integrates the rule-based and neural-based approach which is result-oriented and gaining significance these days.
- As source language, Sanskrit is a linguistically rich language having the credit of old scriptures like Vedas which are not usually accessible and understandable by the general people. These scriptures can be easily understood in other languages through SHH-MTS.
- The proposed system is also beneficial in the teaching-learning process by providing with Sanskrit content. The system helps the students by providing illustrations of grammatical information for the Sanskrit text such as parts of speech tagging, parsing, Sadhi-splitting, word sense disambiguation, and relations between different words of a sentence.
- The system can be readily updated with recurrent frequent updates of performance in terms of corpus, domain, algorithm and rules.
- The SHH-MTS is deployed on the cloud and provided as a service. Its deployment makes it easier to perform and easy to use by the common user as no pre-requisite or knowledge of NLP is required.
- Auto-tuning for neural-based MT used in the proposed system is not possible at the local host due to memory issues, but it is possible on the cloud. Several layers are added automatically to attain maximum accuracy and high speed.

The flowchart shown in Figure 4.1 provides the details about all the development phases of SHH-MTS i.e different linguistic tools output feed for embedding as features in the encoder of Neural-based Encoder-Decoder architecture which trains the systems for learning and predicting the translation of Sanskrit words into Hindi words using linguistic features.

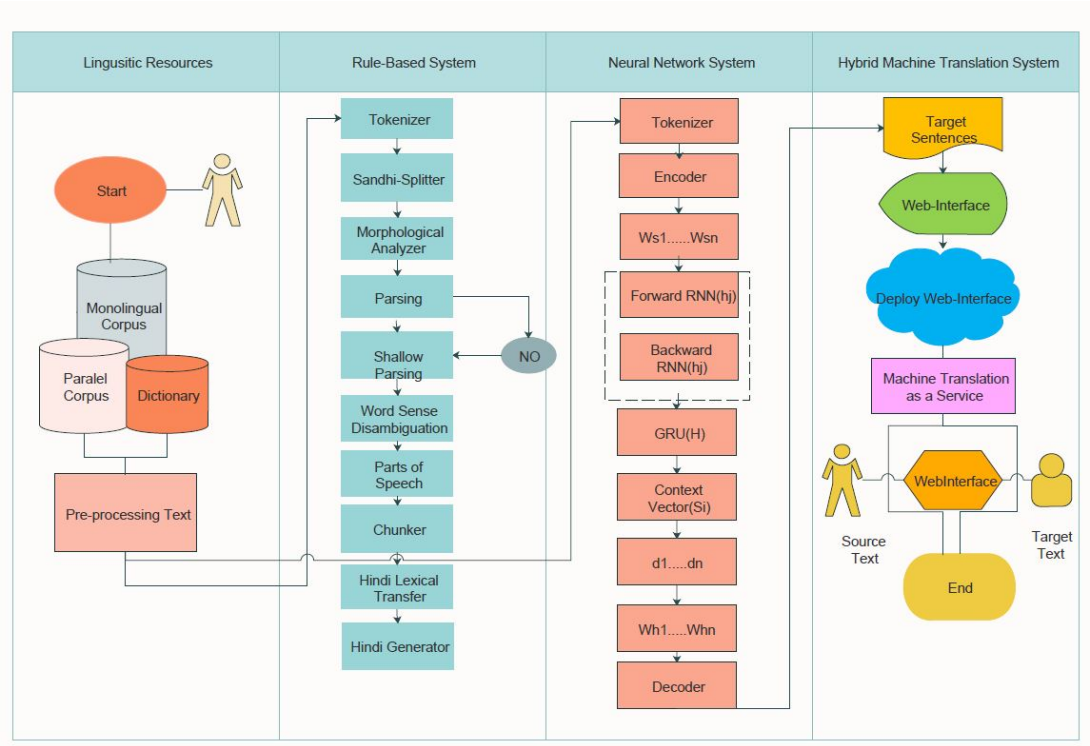


Figure 4.1: Flow Chart of the Proposed System

4.2 Deployment of SHH-MTS on Cloud

MTS is composed of multiple heterogeneous modules having dependencies according to the task performed. It is not easy to resolve these complexities; it is a time-consuming task also. Increasing demand in the request for MTS hinders the performance of the systems. It slows down the response and requires more resources to provide greater computation. This leads to an increase in the computational cost for most of the enterprises and academic institutions. Innovations are required to ride the inevitable tide of change. Recurrent changes in terms of either corpus, domain inclusion, the algorithm of modules, modifying rules or a combination of these lead to improve the accuracy, quality and performance of MT systems. Most of the developments are striving to reduce their computing cost through the means of virtualization. This demand of reducing the computing cost has led to the innovation of Cloud Computing[249]. It offers better computing through improved utilization, reduced administration and infrastructure costs. It is a term used to describe both a platform and type of application. As a platform, it supplies, configures and reconfigures servers, while the servers can be physical machines or virtual machines. On

the other hand, as applications that can be extended to be accessible through the internet; and for this purpose, large data centres and powerful servers are used to host the web applications and web services. The proposed MTS has been designed and developed as a web application hosted on cloud to provide SHH-MTS as a service. The significant

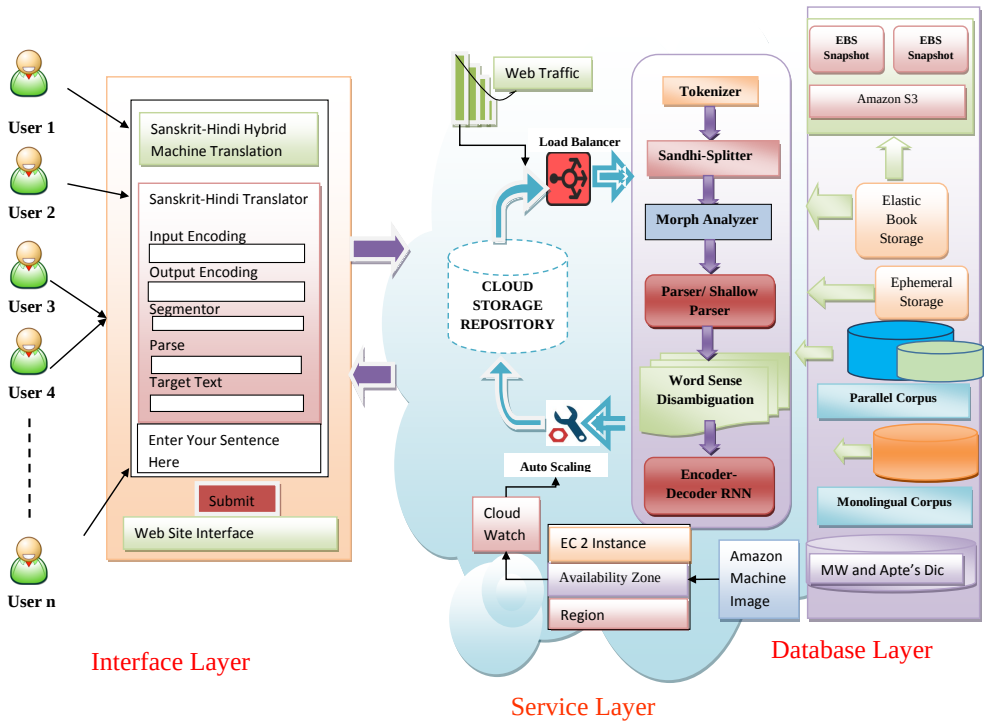


Figure 4.2: Architecture for the Proposed System

features of MTS deployed on the cloud are listed below:

- It provides scalability at the same cost.
- Reduces load by the distribution of task to different servers.
- Fast processing speed due to virtual machines
- Handling of frequent updates in the algorithm, rules, corpus, dictionary and domain inclusion.
- Enhancement, adaptability and scalability are easier to perform.
- Easy to use by the common user as no pre-requisite or knowledge of NLP is required.

These significant features are adopted by several researchers to develop MTS, and by using MTS-as-a- service higher performance has been achieved. Nowadays different cloud service providers are assisting the industries as well as for personal use. Amazon Web Services

(AWS)[250] is one of the cloud service providers which is a secure cloud services platform offering compute power, database storage, content delivery and other functionality to help system scale-up and grow. Therefore, the model can be accessed and utilized according to the need.

Virtualization[251] is a viable option by making an application function as a repository. The key benefits of developing virtual appliance are fine granularity with reducing time for adding and removing computational resources. It would also increase the mobility of application and reduce deployment time. The deployment of virtualization can be performed on the cloud as well as stand-alone. An SHH-MTS is deployed on the cloud architecture as shown in Figure 4.2. The proposed architecture has been divided into three layers, i.e., an interface layer, service layer, and database layer. The interface layer is used for user interaction. In the proposed architecture, the interface has been built in the form of a website for SHH-MTS, delivering Sanskrit-Hindi translation as a service to the users. All the users requesting for Sanskrit-Hindi translation can access the system through the

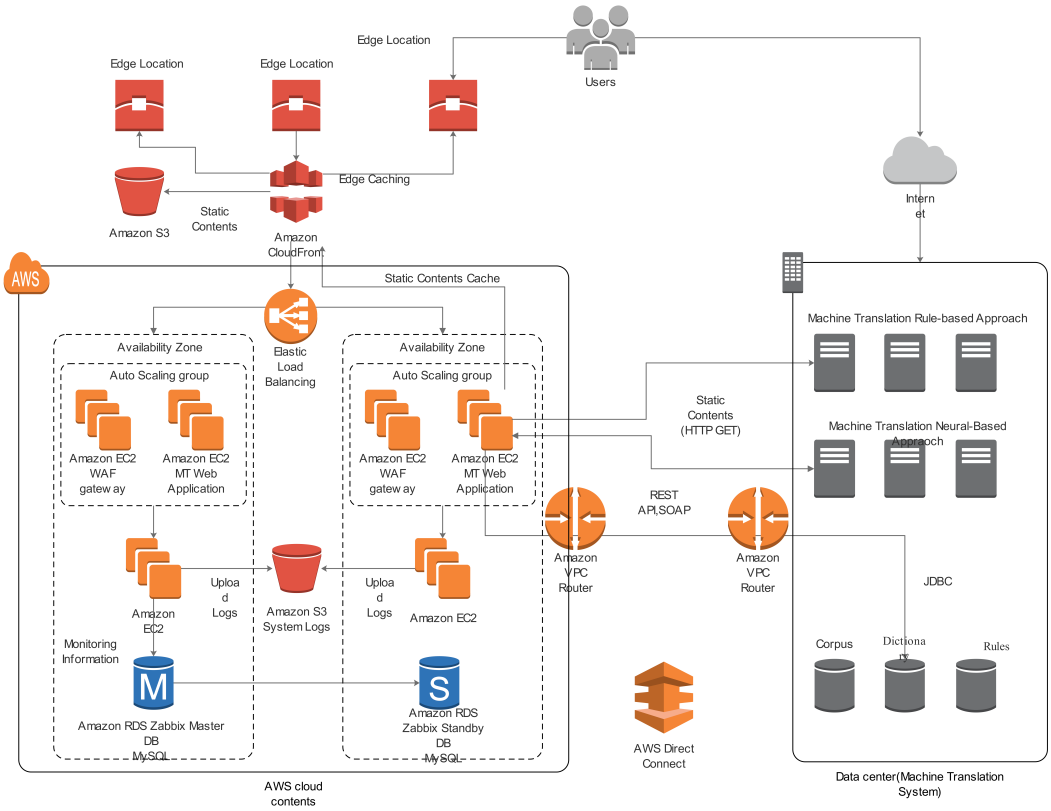


Figure 4.3: AWS Infrastructure used for Deploying SHH-MTS as a Service

interface of the website which is deployed at the service layer. At the back end of the translation system, the output is generated by various linguistic tools and encoder-decoder architecture of RNN. All these user requests are stored in the cloud storage repository and forwarded further for processing into the SHH-MTS. Even load balancing and auto-scaling are performed on this layer to handle the traffic demands. It also manages to

address advance routing needs by dynamically scaling the web application to changing traffic on demand. It can create capacity groups of servers that work accordingly to the demands. The AWS Elastic Compute Cloud (EC2) allows users to use virtual machines of different configurations as per requirements. It provides a more secure model for every host. The database layer is comprised of the parallel corpus, monolingual corpus, and dictionaries. It is used for the physical interface between the application and the database. Simple storage service(S3) is used to allow users to store and retrieve various types of data API calls. With provisioning of additional hardware resources, it is possible to keep the response time within optimum limits as the load increases, but this increases the cost.
Thus, the objective of this proposed work is to deploy MTS on the cloud with provisioning of larger computation resources. It will help to increase the scalability of the system and improve the response time. Cloud deployment requires optimum resource utilization which is possible only when an application can scale up and scale down rapidly. This is easily feasible for the proposed SHH-MTS to scale-up or scales down in real-time. In this work, a hybrid MTS model has been deployed on the cloud of Amazon(EC2) as depicted in Figure 4.3 having better accuracy, CPU utilization, and minimum response time. It also eases the deployment and scope of extension or manageability due to which the performance of the proposed system is better than a standalone version of systems.

4.3 Experimental Details

The experiment has been conducted on the developed SHH-MTS to measure various QoS parameters. The technical specifications along with its versions are displayed in Table 4.3. For the deployment of SHH-MTS on cloud infrastructure, Amazon Elastic Computer Cloud(EC2) has been used. It simulates scalable deployment of MTS by providing a web service. It provides secure, resizable compute capacity through which booted an Amazon Machine Image(AMI) 3.0 for deep learning to configure a Virtual Machine(VM), also called an Instance. The developed system includes dictionaries, parallel corpus, monolingual corpus, program codes, algorithms, lexical resources, rule database, machine-learned data and its models; and it is packed as AMI. The experiment conducted was based on Linux Kernel version 3.4.34 Operating system through which user program interacts with the kernel. Virtualization was first performed with Xen and later with Compute Optimized C5 instance, which was based on custom architecture around Kernel-based Virtual Machine(KVM) hypervisor. The processor was ported with IA-64. Elastic Block Store (EBS) volume as a rooted device was used for storage with G3.4*large type. It provides raw block devices that can be attached to the Amazon EC2 instance. It also supports cloning and snapshot. So, the system image has been cloned to other virtual machines. It is built on replicating storage when the failure of one component would not cause data loss. EBS volumes can be attached or detached from Instance or VM while

they are running and moved from one VM to another VM. A Simple Storage(S3) is also used which has been accessed (read and write) through the API. The experiment was based on 16 core processor and 122 GB RAM. Rule-based auto-scaling has been used which adapts according to the CPU utilization threshold though it takes several seconds to scale up and scale down. Although the VM start-up time is not dependent on VM type, AMI size and data centre location, etc still it takes a few seconds to configure.

Table 4.1: Technical Specifications along with Versions

Technical Specifications	Version/Number
Amazon Machine Image(AMI)	3.0
Linux Kernel O.S	3.4.34
Compute-Intensive Virtual Machine	C5
Elastic Block Storage(EBS)	G3.4*large type
16-core Processor	IA-64
RAM	122 GB
Virtual Machines	(4,8,12,16)
GPU	Geforce GTX 980
Memory(for each instance)	61 GB
Elastic Compute Unit	12
CPU	2 (for each VM)
Cache	4MB

Each virtual machine has a GPU with 122 GB of RAM and 16-core processor to achieve a high throughput speed approximately 2500 words per second. This speed is not possible for normal systems because in it one epoch will take approximately two hours to run. As many as 5 nodes have been allocated for this experiment. On each node, SHH-MTS has been pre-installed with the help of AMI of 1GB size. It takes 60 seconds to boot. For each created instance, there have been 12 elastic computing units and 4 virtual CPUs, and a memory of 61 GB for a single instance (EBS storage only) with high network performance. After all the environment set-up, the system gets ready to run and performs the performance testing. The system first runs rule-based linguistic tools, which give its output to CSV file. This file is then processed on TensorFlow with Keras with python 3. It makes an evaluation on the basis of parameters such as average response time, cost optimization, throughput, server load, total time taken concerning rule matching probability. The number of computational resources (Storage, O.S and Instance type), data, virtual machines vary across to evaluate the performance of MTS. The interface designed and hosted on the cloud has been displayed in Figure 4.4.

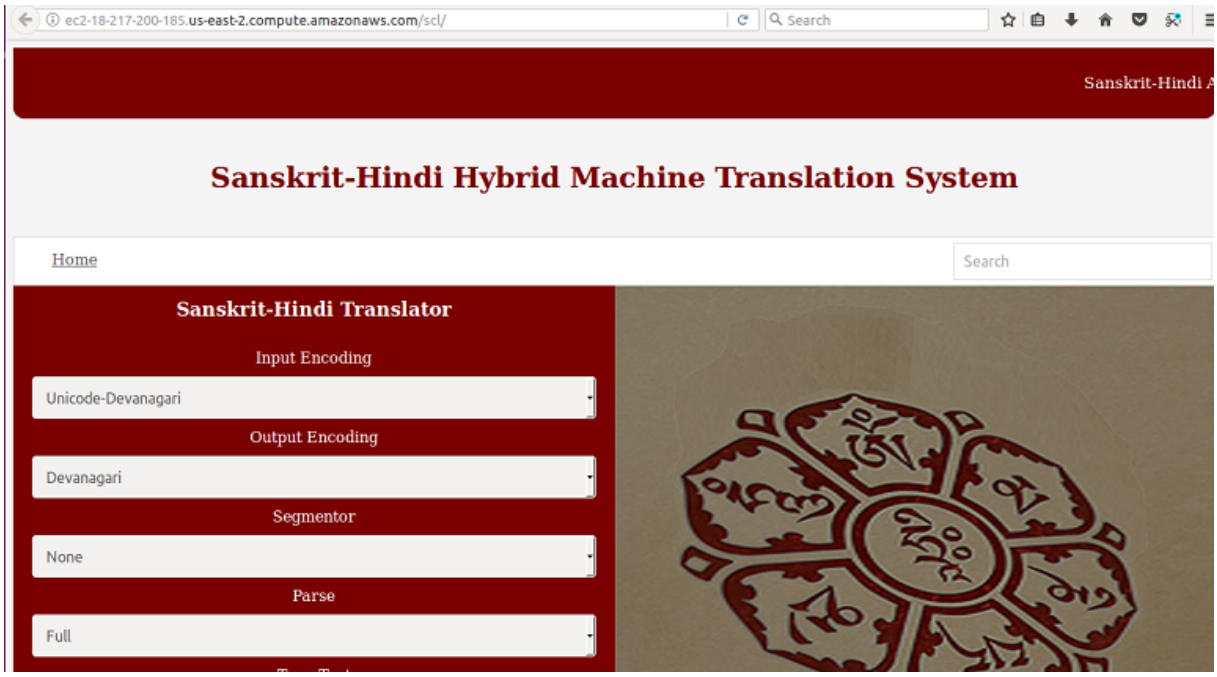


Figure 4.4: SHH-MTS as a Service

4.4 Validation of Results for the Proposed System on Cloud

The performance of the proposed work was evaluated by conducting a diverse statistical test. This section has been divided into two subsections. The first section compares the proposed SHH-MTS as a service with the existing works. While the other subsection focuses on the performance of MTS based on different approaches, i.e., rule-based, neural-based and hybrid. As the proposed system was built in iterations, i.e rule-based approach was applied first. Then, the neural-based approach, and finally, the output of rule-based was fed as features to neural-based forming it as a hybrid approach. Different statistical tests were conducted for evaluating the performance of the proposed system in terms of average response time to rule-matching probability, auto-tuning process, cost optimization, throughput, server load and total time taken to the number of virtual machines. The results were achieved by implementing multiple runs which manifested multiple units of values at different intervals of time.

4.4.1 Comparative Analysis of the Proposed Work with Earlier Research Work on Cloud

Throughput is directly proportional to the number of processes completed; and it is also used to assess the performance of the translation system. It is calculated on the basis of resources used and time consumed.

- *Throughput on Stand-alone MTS*: Its calculation is done by executing a whole book of sentences using 16 core processor, 122 GB RAM, EBS only instances and G3.4x large type. The same job is divided into several tasks to execute on different computing resources. Table 4.2 highlights the results of the standalone system of the earlier proposed work[205] on Hindi to Punjabi Cloud-based Sampark MTS tested on dual-core CPU and 1 GB of RAM, and the proposed SHH-MTS as a Service.

Table 4.2: Throughput Results on Standalone

Book	**Total Time Taken on Stand-alone**
Nirmal by Premchand[206]	198min+13sec
Sankshepa RaamaayaNam (on SHH-MTS)	130min+50sec

- *Throughput on Virtual Machine*: First of all, 2 virtual machines were allocated which took 5440 sec to process. Later on, experimention with 4 virtual machines which took 2113 sec. The time for processing reduced rapidly on using more number of virtual machines as it took 790 sec for 8 virtual machines and 320 sec for 12 virtual machines. The comparison made with CBSMTS showed that it took 970sec on 12 virtual machines as depicted in Table 4.3.

Table 4.3: Throughput Calculation on Virtual Machines

Number of VM	**Throughput on CBSHH-MTS**	**Throughput on [206]**
2	90min+ 40 sec	112min+19sec
4	35min+ 60sec	54min+6sec
8	13 min +13 sec	26min+26sec
12	5 min+20 sec	16min+10sec

- *Computing Nodes Reducing the Time for Computation*: As shown in Figures 4.5 and 4.6, a test was performed for sentences of different length, i.e., 20,50,100,150,200 and 250; and varying computing nodes 2,3,4,5,6 and 7. The results indicate that an increase in the nodes results in reducing the time.

- *Response Time*: The response time is the time required for output sentence after passing the input sentence to MTS. The time taken for the proposed CBSHH-MTS was lesser as compared to the previous research work for sentences of different length i.e.,20,50,100,150,200,250.

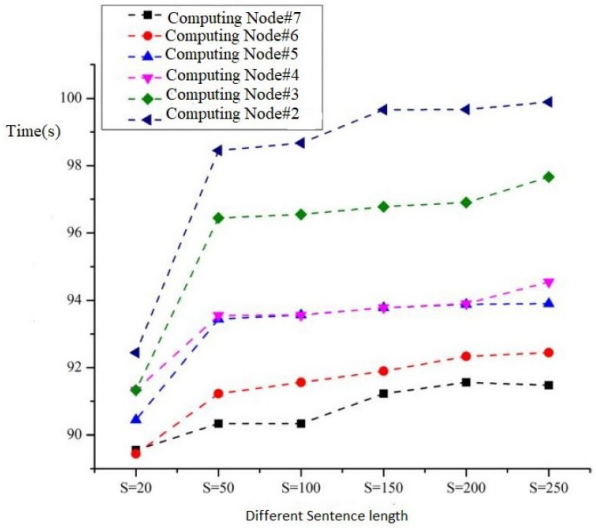


Figure 4.5: Deployment and Usage of Cloud Infrastructure [206]

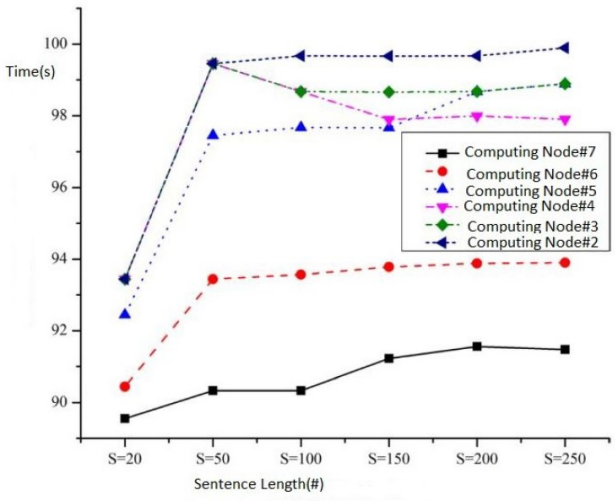


Figure 4.6: Deployment and Usage of Cloud Infrastructure for SHH-MTS as a Service

4.4.2 Performance Analysis on Deployment of Rule-based MT, Neural MT and Hybrid MT on Cloud

An elaborate description of the rule-based approach followed in the development of proposed system is given in Section 3.1, neural-based approach in Section 3.2, and that of hybrid in Section 3.3. These have been deployed on cloud seperately, and their performance has been analyzed on the basis of various parameters. The average response time which depends on the matching of rules with the rule database is depicted in Figure 4.7. While the average response time on the number of match action rules is displayed in Figure 4.8. Further, the CPU utilization on cloud-based is on the packet arrival rate which is shown in Figure 4.9. Though the AWS allows to pay as you use service, the cost according to resources required is displayed in Figure 4.10. The throughput is directly proportional

to the number of processes completed and calculated based on resources used and time consumed as shown in Figure 4.11. The throughput increases with an increase in the number of virtual machines as in Figure 4.12. Figure 4.13 highlights that the hybrid approach decreases the server load and it is more as compared to rule-based and neural-based approaches. The auto-tuning process (automatic addition of hidden layers) leads to reduce this load significantly for both systems after the deployment of the proposed system on the cloud. The graphs depict that the performance of a hybrid system, is better than the rule-based and neural-based approaches.

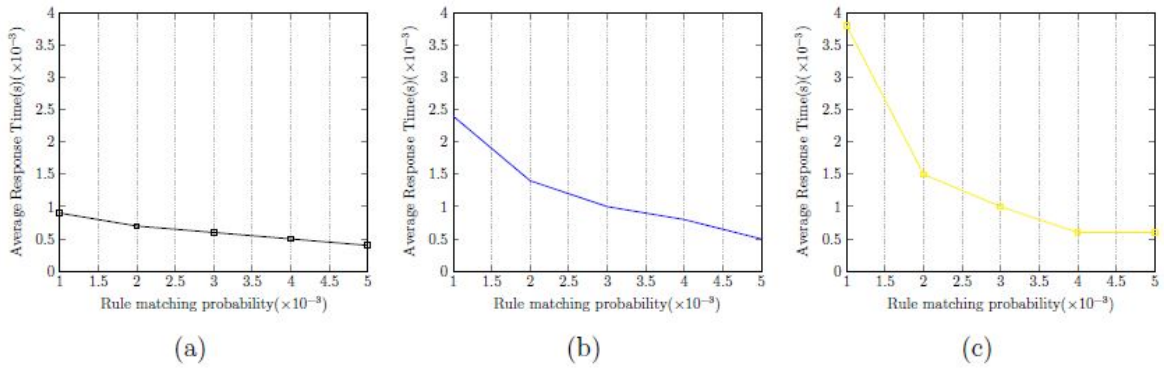


Figure 4.7: Average Response Time pertaining to Rule Matching Probability (a) Hybrid MTS, (b) Neural-based MTS, and (c) Rule-based MTS

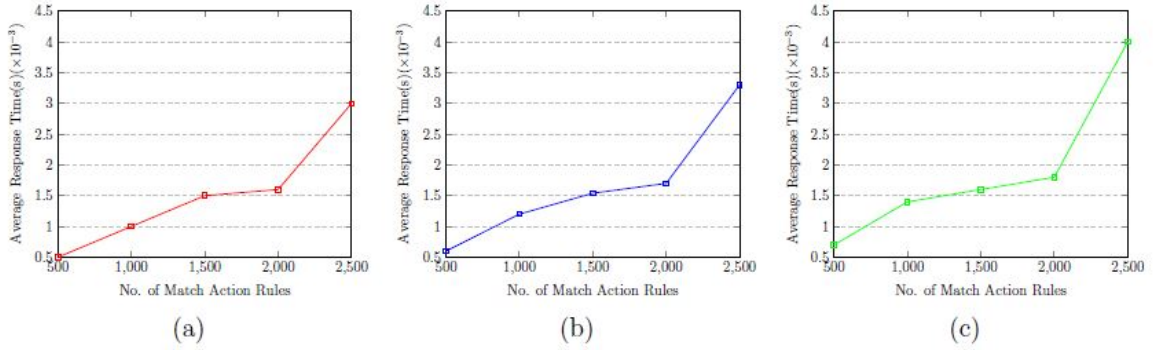


Figure 4.8: Average Response Time pertaining to Number of Matching Action Rules (a) Hybrid MTS, (b) Neural-based MTS, and (c) Rule-based MTS

4.5 Conclusion

The hybrid machine translation system developed for converting Sanskrit to the Hindi language integrates the rule-based and neural-based approach which is result-oriented and gaining significance these days. It is a complex application with a large number of heterogeneous modules. Deploying such a complex application on a stand alone machine becomes a difficult and time-consuming task. The existing Sanskrit-Hindi MTS has many

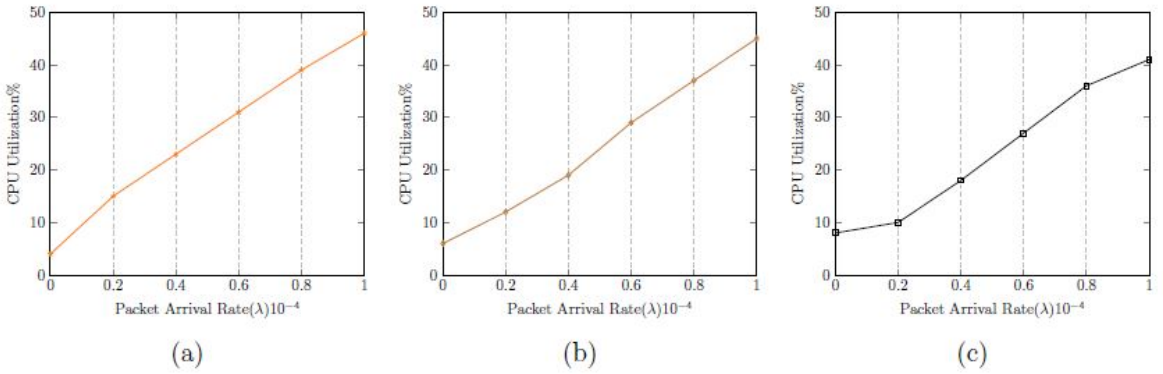


Figure 4.9: CPU Utilization pertaining to Packet Arrival Rate (a) Hybrid MTS, (b) Neural-based MTS, and (c) Rule-based MTS

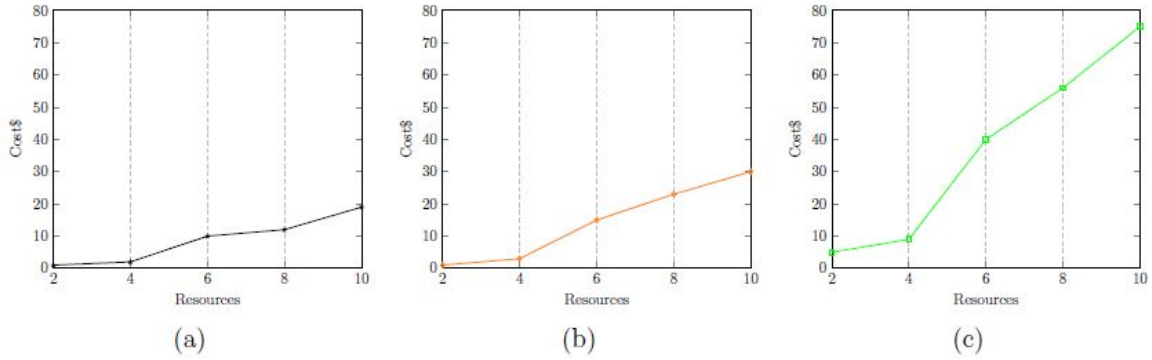


Figure 4.10: Cost pertaining to Resources (a) Hybrid MTS, (b) Neural-based MTS and (c) Rule-based MTS

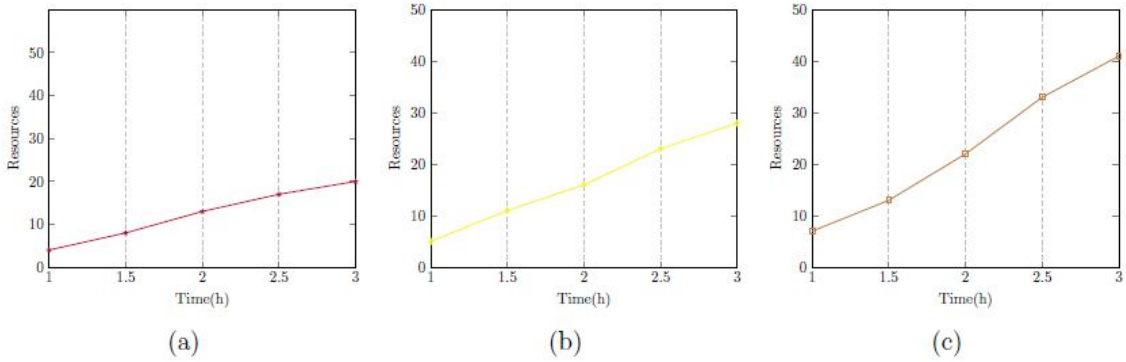


Figure 4.11: Throughput (a) Hybrid MTS, (b) Neural-based MTS, and (c) Rule-based MTS

drawbacks such as slow speed, less data accuracy, and low response time. All these factors adversely affect the performance of the system. It has been observed that local server takes more time to respond and provides lesser accuracy. Therefore, offering MTS as a cloud service is a better proposition for increasing its performance in terms of accuracy

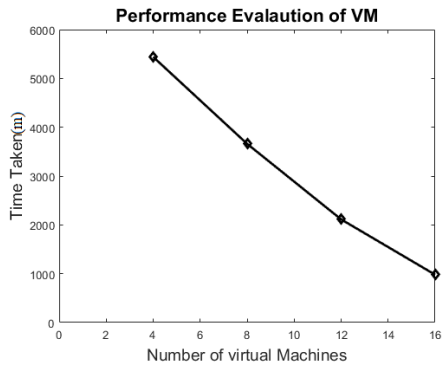


Figure 4.12: Time Taken with respect to Number of Virtual Machines

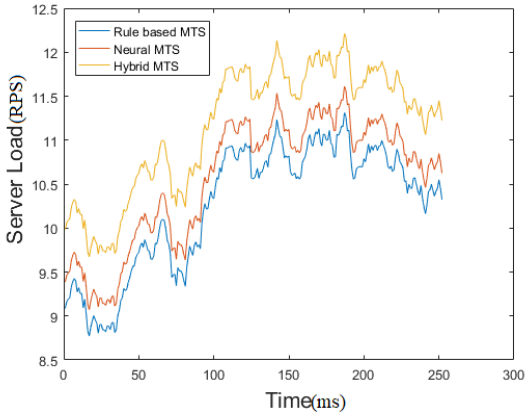


Figure 4.13: Server Load with respect to Time

and response time. Moreover, auto-tuning for neural-based MT is not possible at the local host due to memory issues, but it is possible on the cloud. Several layers are added automatically to attain maximum accuracy and high speed. The proposed CBSHH-MTS provides better throughput, rule matching probability and number of matching rules in comparison to the stand alone systems.

The next chapter validates the proposed work with a case-study. It outlines the developed taxonomy of error analysis based on different linguistic levels, i.e., orthography, morphology, lexical, syntax, semantics and pragmatics.

Chapter 5

Case Study on Proposed Machine Translation System

The previous chapter presents the description for the deployment of the proposed work on the cloud. It offers translation as a cloud service by improving the quality of service (QoS) from the stand-alone system.

This current chapter outlines the developed taxonomy of error analysis based on different linguistic levels, i.e., orthography, morphology, lexical, syntax, semantics and pragmatics. Consequently, the previous taxonomies were expanded to adapt the errors transpired in morphological rich Indo-European languages. As far as direct access to Sanskrit text is concerned, it requires good grammatical knowledge, manual access to the dictionary, knowledge of syntax and semantics which is a tough and time-consuming process. This interactive interface will assist the school as well as university students enrolled in distance education by promoting self-learning. The main objective of the proposed system is to make the scriptures and philosophical texts such as Gita, Ramayana and Upanishads, available in the Sanskrit language, accessible to the common user.

The chapter presents the proposed error taxonomy in Section 6.1. The case study of the developed system for Sanskrit to Hindi translates is performed across the proposed error taxonomy in Section 6.2. Lastly, the chapter concludes in Section 6.3.

5.1 Proposed Error Taxonomy

In morphologically rich languages, grammatical relations indicates by changes in the word. One word can take multiple forms, making it harder to map the dictionary meaning and grammatical rules. The automatic evaluation measures developed are too coarse into such case thus, human evaluation indulged. Though human evaluation measure is quite generic and involves qualitative analysis such as fluency or adequacy, it neither captures nor quantifies word-level errors.This work, provides a baseline to the community with a classified taxonomy for the error identified and their resolution in the translation. This will eventually improve the MT system accuracy.
The proposed framework of linguistically motivated error taxonomy for morphologically rich Indo-European languages is shown in Figure 5.1 Error recognition is not an easy task. All the errors incurred are not easy to find as some of them are infused in the sentence throughout. Identification of an error is a complex task in building the classification taxonomy of errors. In this work, the errors encountered in different corpora, as shown in Table 5.1, have been classified on the basis of linguistic categories such as orthography, Morphology, Lexis, Grammar, Syntax, Semantics and Pragmatics.These have also been exhibited in Figure 5.2. This broad classification of categories indicates the linguistic level at which the errors were identified. These have been described in detail in the following subsections. All these errors relate to Sanskrit to Hindi translation applied on various corpora. This error taxonomy is quite helpful to optimise the performance of MTS.

5.1.1 Orthography

The orthography level of language refers to the agreement of writing the language. It includes rules of transliteration, spelling, capitalization, compound splitting, tokenization, and punctuation. The transliteration maps words from one language to another using the phonetic base[252]. The orthographic errors have been classified into transliteration, spelling, capitalization, compound splitting, tokenisation and punctuation.

- The transliteration involves the conversion of text string from one orthography to another such as proper names, numbers and punctuation. This conversion is language independent. As addressed by [253], proper names which are to be transliterated are out-of-vocabulary.

- Spelling is required to be addressed as a minor mistake in spelling or typo-error may result in out-of-vocabulary words in training data as addressed by [254]. There are various statistical and rule-based methodologies proposed by [138] to overcome these orthographic errors.

- Capitalization requires to be addressed as the capital text may lead to transliteration

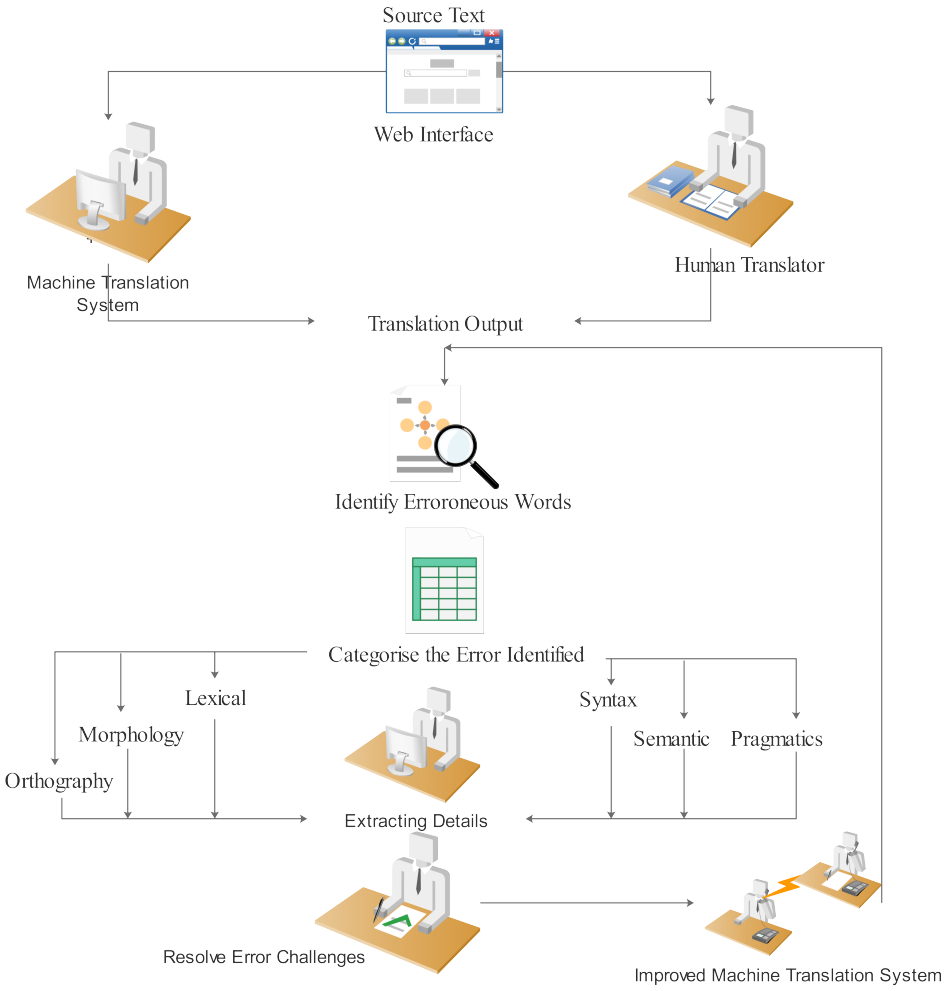


Figure 5.1: Framework of Linguistically Motivated Error Taxonomy for Morphologically Rich Indo-European Languages

rather than translation. Therefore, all the text needs to be converted from upper case to lower case letter. Several experiments have been performed [255] quantifying the capitalization error in the translation process [256]. [257] proposed handling true casing with the Hidden Markov Model(HMM) requiring pre-processing and post-processing. The proposed models replaced the word with its most frequent form of [258]. There is also a bilingual capitalization model for true casing the output using probabilistic approach[255] and using random field[259].

- Compound Splitting[260] plays a significant role in Sanskrit language processing as it is composed of Sandhi formed words. It is the separation of compound words into their original constituents, otherwise, the entire meaning of the sentence will be changed.[215, 222, 261, 262]

- Tokenization is converting the string of words into tokens. This step is the pre-

processing of MTS.

- Punctuation's misuse causes an error when performing the translation of the sentence. The different punctuation marks such as full stop, comma, sign of exclamation placed in the wrong position will reflect the change in meaning and leads to ungrammatical sentence.

5.1.2 Morphology

It is the study of the structure of language morpheme for a language such as an affix or stem[263]. It imposes a challenge in translation in the case of rich morphological language translation pairs.The morphemes provide syntactic information such as tense, gender, count, case, etc.The morphologically rich language has many different surface forms which lead to different meanings and cause different types of errors as listed below:

- *Inflections*: Inflectional morphology involves different word forms in certain grammatical categories covering affixation and vowel change. In this proposed work, the morph analyzer gives all possible inflexion forms such as the pratipadikas from the Monier Williams dictionary. If a word inflexion form is not provided from the dictionary, it can lead to a wrong word meaning in the output.

- *Derivational*: It provides the morphology with a type of word formation that creates new lexemes, either by changing syntactic category or by adding substantial new meaning (or both) to a free or bound base. In our proposed work, it supplies the analysis of words after adding the derivational morph output if the pratipadika is derivational.For instance, the krt pratipadikas and taddhita pratipadikas are analysed further for krt/taddhita analysis in the derivational morph analysis.

- *Parsing*: It provides an analysis of sentence words into its constituents, resulting in a parse tree showing their syntactic relation to each other, which may also contain semantic and other information. For the purpose of the proposed system, the parser takes the input of the morph layer and produces the correct morph analysis in the context along with the karaka analysis.

- *Unrecognized Words*: Some of the words in the input sentence are not recognized by the morphological analysis. Since the output of later modules depends on the output of the morphological analyser heavily, a provision has been made in the proposed work to supply an analysis of unrecognised words manually. This analysis is made by the user, and then used by later modules.

- *Pruning*: This category of morph analysis assists the morph output by providing answers to rare words. In the proposed system, the results have been further pruned to provide the answers for the rare pratipadikas as these are not found in Apte's dictionary.

5.1.3 Lexical

In this category, all the errors affecting the lexical items have been considered. It considers a word as a whole instead of an error in the character of the word. Based on the type of words, i.e., content words carry the content or meaning, while function words construct the grammatical relationship with other words in the sentences.The errors based on their categorization are listed as below:

- *Omission*: This type of error is encountered when the translation of a word present in the source text is missing in target translation.
- *Addition*: This type of error arises when the word in the source text is not present, but is added to the translation text.
- *Untranslated*: Apart from the omission and addition for some words, the machine translation engine cannot find a translation for a given source word. In such cases, the solution is to copy the source word as it is in target translation.
- *Unknown Words*: There are few unknown words where it is difficult to find translation for such words. Such type of errors can be dealt with external resources or addition of manual word sense disambiguation rule in the database[264] of the MTS.
- *Spurious Words*: These words do not have a counterpart in another language. The Machine translation system should be able to identify such words. It can be dealt with diverse solutions such as omitting a spurious word from source sentence or using an alignment procedure.

5.1.4 Syntax

It refers to the rules for constructing the sentences in natural language. Each language has its syntax defined by the rules.The languages are categorised into free word order and fixed word order.

- *Reordering*: If the source sentence is having different syntax, then the target sentence in fixed word order can lead to error requiring reordering of the sentence. There are different reordering algorithms built for reconstructing the syntax of the sentence. It can also be done by modifying the parse of the sentence.

5.1.4.1 Semantics

It relates to the meaning of words and phrases, and the combination of both. It can lead to errors such as sense disambiguation, wrong choices, collocational errors, identifying multiword expression, idioms and handling tense aspect modality of the sentence.

- *Sense Disambiguation*: It causes a problem when a word is translated into its possible meaning, but it is not correct according to the given context which leads to a wrong translation.

- *Wrong Choices*: It occurs when a wrong word is chosen without apparent relation to translate the given source word into the target word.

- *Collocational Errors*: These errors occur when the other word normally accompanies it; hence, this error takes place for a block of words. Their selection is not semantically motivated which causes the error. Though the collocational errors can be considered as instantiation of the previous error, but these can also be considered separately.

- *Multiword Expression*: There are certain words used as expressions which are made up of at least two words and which can be syntactically and/or semantically idiosyncratic. Moreover, they are not identified as a single unit and may cause an error at a certain linguistic level.

- *Idioms*: These are complex constructions of language used creatively across almost all the text genres. Idioms pose problems to natural language processing (NLP) systems due to their non-compositional nature; and the correct processing of idioms can improve a wide range of NLP systems.

- *Tense Aspect Modality*: All languages mark tense, aspect and modality (TAM) in some way, but the markers don't have a one-to-one mapping across languages. Many errors in machine translation (MT) occur due to wrong translation of TAM markers. The curtailment of such error can lead to improve the performance of an MT system.

5.1.5 Pragmatics

It is the deepest level of computational linguistics. The errors occuring at this level are discursive options that are not most expected. There are different situations at this level such as style, variety words that should not be translated, an agreement between noun and adjective, and Vibhakti.

- *Style*: These errors occur when a bad stylistic choice of word is used in translating a sentence. For instance, repetition of the word in a nearby context, where synonyms would be better to be substituted.

- *Variety*: These errors cover the cases when the target of translation is a certain language, but instead lexical or grammatical structure from a variety of that language are used.

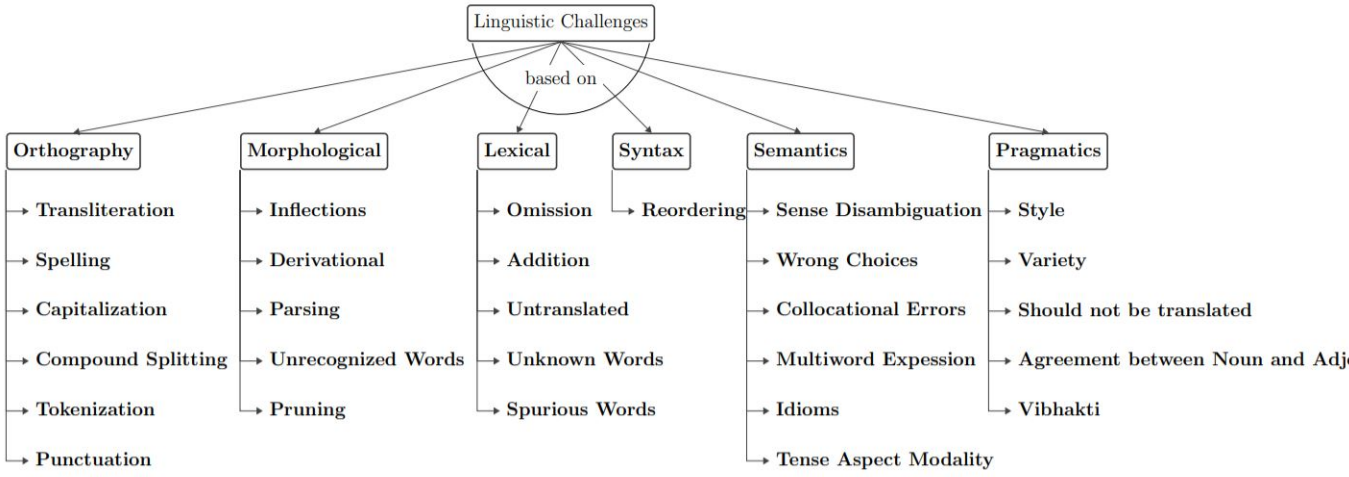


Figure 5.2: Proposed Taxonomy

- *Should not be Translated*: This category considers all the word sequences in the source language that should not be translated in the target language. For instance, a book name is translated from source to target language, and it results in adding an error.
- *Agreement between Noun and Adjective*: In the target language translation generation, this involves an agreement between noun and adjective such as adding 'ne', dropping 'ko' at unnecessary places in the sentence.
- *Vibhakti*: Vibhakti provides information on respective karaka. Vibhakti guides for making a sentence in Sanskrit. In Sanskrit, there are seven types of vibhakti, i.e nominative, instrumental, dative, ablative, genitive and locative. Karaka theory acts as a media between grammar and reality. For example, handling Kriyamula Vibhakti and agreement, handling agreement for Sasti Vibhaki, Karta and Karta Samanadhikarana, and final agreement between the noun and verb.

5.2 A Case Study of Proposed Error Taxonomy on Sanskrit to Hindi Language

This section is comprised of the peculiarity of the case study performed for Sanskrit-Hindi language. In the following subsections, the translation system engaged for this particular case study has been explained. Further, the results obtained from both automatic and manual metrics have been analysed. Finally, the errors across the proposed taxonomy in the linguistic levels by error identification and their classification have been mapped.

5.2.1 Prerequisite

Before we start analysing our results, the following prerequisites are supposed to be considered:

1. The results have been presented as the number of errors present per dataset, but exception persists in certain cases where a word contains two errors. The words with two errors have been set into the total number of errors.

2. The comparison of the system based on error has been performed only on the lexical level. The performance of the system acts as an indicator to calculate its probability.

The proposed MTS model is capable of covering all types of error arising out of rule and neural-based approaches.

5.2.2 Error Identification and Classification

An attempt was made to investigate how the error occurred in all the three systems across different corpora. Step-wise analysis was undertaken. Firstly, the number of correct words and errors over every corpus were identified. Then, the identified errors were classified into orthographic, morphology, lexical, grammar, syntax, semantic and pragmatics. The probability of these errors was calculated and presented in Table 5.1.

This work proposes a teaching and learning framework for accessing Hindi from the Sanskrit language. It uses Anusaaraka (Available at:https://ltrc.iiit.ac.in/Anusaaraka), a language accessor platform, Sanskrit Computational Linguistic tool (Available at: http://sanskrit.uohyd.ac.in/scl), heritage site and Sanskrit digital text(available online at:http://sanskrit.inria.fr) to build the system. This system can be effectively used as a replacement to standard ancient teaching methods. The several features or benefits of the proposed work are elaborated as below:

5.2.3 Features of the Proposed Teaching-Learning Framework

- The essential step in teaching and learning Sanskrit is knowledge of grammar, script, vocabulary, syntax, phonetic repository, grammatical rules such as sandhi, morphology (derivational and inflectional) and prosody along with literature. Such information derives the structural and semantic sense of the sentence. The statistical analysis made available by the system assists in deciding the aspect to be covered in teaching a text by the teacher. The graphical interface of the system is meant to enhance the interest and attention of the students and teachers.

- It support the linguistic analysis undertaken with respect to morphology, parsing, sandhi-splitting, etymology, samasa, karaka, segmentation and dictionary.

Table 5.1: Error Identification and Classification Across Different Sanskrit Corpora

Corpus	Lexical Richness(%)	Avg. phrase Length(%)	Correct(No.)	Error: lexical(No.)	Error: Semantic(No.)	Error: Grammar(No.)	Error:Orthography(No.)	Error:Syntax(No.)	Error:Pragmatics(No.)	Error: Morphology(No.)	Hit rate(%)	Lexical hit rate(%)
Rasaratnasamuccayatīkā	0.18	9.8	9986	800	8	357	299	1	234	0.85	0.93	0.92
Rasamañjarī	0.22	7.2	10550	694	5	576	244	0	281	232	0.85	0.94
Bhavaprakāśā	0.25	7.0	11096	878	2	314	280	0	161	213	0.87	0.93
Laṅkāvatārasūtra	0.16	11.6	9305	1730	17	347	254	3	388	421	0.77	0.85
Bodhicaryavatāra	0.25	7.7	10845	922	4	555	337	0	244	214	0.84	0.92
Yājñavalkyasmrti	0.27	7.0	11764	1059	19	557	234	1	150	162	0.85	0.92
Gokarṇapurāṇsārah	0.19	6.6	12497	1100	18	431	220	4	144	142	0.86	0.92
Sārñgadharasamhitādīpikā	0.17	8.1	12449	833	2	532	372	1	305	165	0.85	0.94
Dašakumāracarita	0.28	16.9	12081	1283	19	439	357	5	252	98	0.83	0.91
Rasaprakāsasudhākara	0.21	7.2	15417	888	9	553	345	0	335	189	0.87	0.949
Visnusmrti	0.24	5.7	15071	1378	29	809	371	4	257	173	0.84	0.92
Buddhacarita	0.20	9.1	15748	1485	22	723	536	5	237	247	0.84	0.92
Mrgendraṭīkā	0.16	12.4	17443	1902	23	705	569	5	478	129	0.82	0.91
Sātvatatantra	0.30	7.1	6240	1259	4	309	143	0	1923	136	0.63	0.87
Arthašāstra	0.29	9.3	8877	962	5	450	318	0	248	156	0.81	0.91
Hitopadeśa	0.23	7.1	9822	900	12	414	167	3	77	199	0.86	0.92
Rasaratnasamuccaya	0.18	7.2	23311	1656	5	1175	672	1	546	175	0.85	0.93
Mugdhāvabodhinī	0.13	12.7	24613	2087	20	1268	797	0	443	214	0.84	0.92
Rasarnāva	0.13	6.8	26198	1895	10	1427	1031	1	711	185	0.83	0.93
Manusmṛti	0.17	7.0	31177	2728	57	1650	890	9	342	157	0.84	0.92
Ayurvedadīpikā	0.12	12.8	31355	2834	51	1259	902	10	545	126	0.84	0.92
Kurmapūranā	0.14	6.6	33308	2856	26	1244	715	17	391	145	0.86	0.92
Bṛhatkathāślokasamgraha	0.12	6.8	49372	5482	114	1997	1401	28	864	156	0.83	0.90
Ānandakanda	0.11	6.9	68767	5896	17	3720	1731	17	1791	146	0.83	0.92

- There are numerous government and non-government organisations providing distance education such as Rāśtriya Sanskrit Sansthān(available online at: http://www.sanskrit.nic.in/) and Samskrita Bharti(available online at: https://www.samskritabharati.in/). The postal notes provided by them are in some cases assisted with instructors in the source language. This system interface would be highly beneficial for all the potential users in understanding and carrying out an in-depth analysis of Sanskrit language in Hindi.

- A trend has been observed among a certain age group of people who have the enclination to understand their culture through the study of religious scriptures such as Geeta and Mahabharata. Although some Sanskrit linguists have translated these scriptures in Hindi, but with their own perspective which may or not match the readers' opinion. This interface provides the intermediate output by giving final translation to the users.

- The ancient methods of learning and teaching Sanskrit require eight to twelve years of dedicated study, making it impossible in today's scenario. It involves learning grammat-

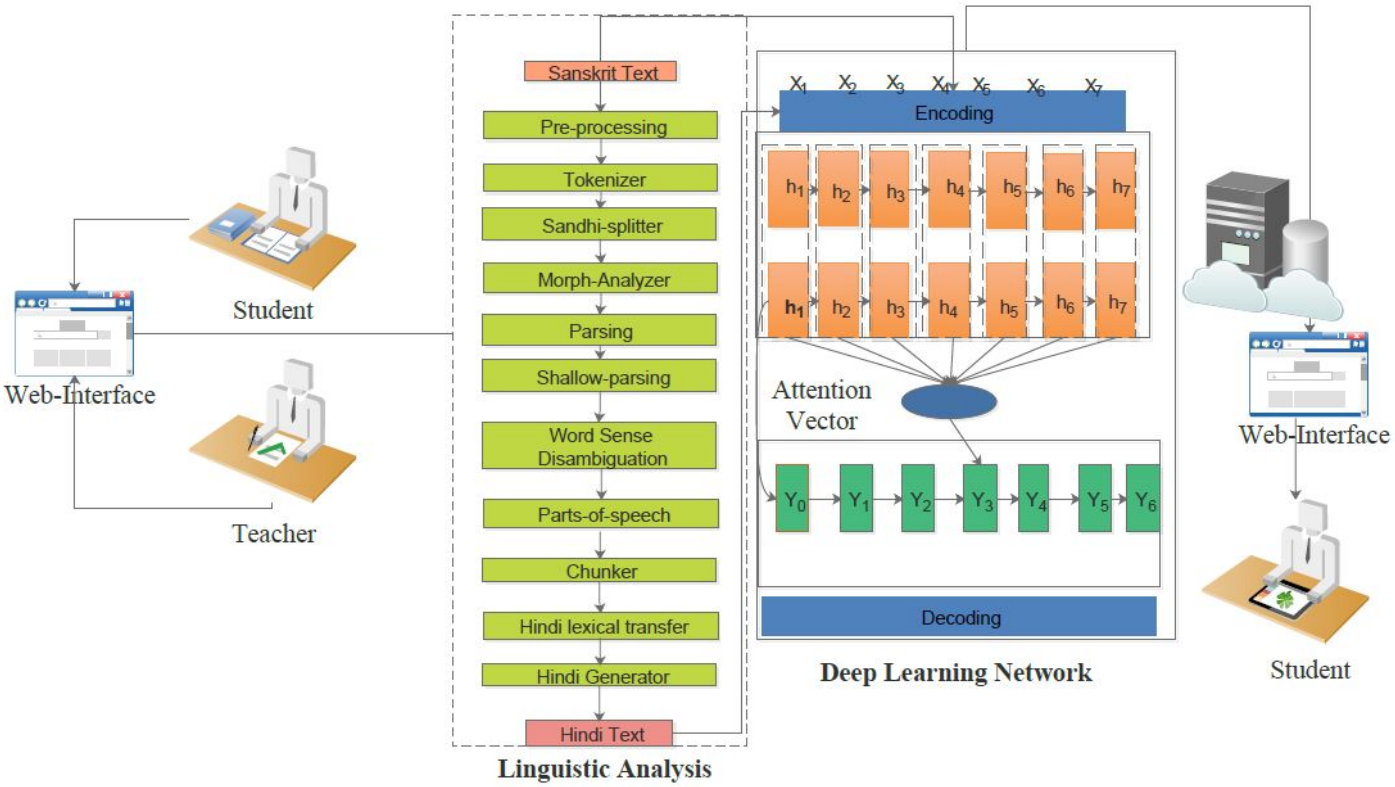


Figure 5.3: Teaching-Learning Framework

ical rules, word-forms memorization, lexical forms, capturing the semantic and syntactic sense of the sentence. In the technology-savvy world, students want an alternative out of this memorization to save time and efforts. In such a situtation, this interface is going to assist the students at one click of a button. It will also promote self-learning.

Figure 5.3 depicts the teaching-learning framework developed through the rule-based and deep learning based Neural Machine Translation(NMT) approaches. The user can interact with the system with the help of web interface developed for a better user-friendly experience. The Sanskrit text follows a sequence of linguistic processing, and then is fed to deep neural network. A detailed explanation has been given in the following sections:

5.2.4 Implementation

The corpus data is found suitable for applying the NMT model. It is a two-fold process, i.e., clean text and split text. The sentences are sorted in a batch based on sentence pair by length, and break similar-length sentences into mini-batches. Therefore, the training corpus is recurrently shuffled, and the corpus is broken into maxi-batches and again splitted in mini-batches. These are processed by applying gradient for parameter update. The proposed work has a pipeline architecture. which takes input from its previous phase and performs computations and passes the output to the next step. Different tools are divided

Table 3: Metric Analysis of Sanskrit to Hindi Translation

Sentence	Sanskrit	Hindi	Reference Trans
1	साध्वोः शीघ्रं मैत्री भवति	साधु जल्दी से दोस्त बन जाते हैं	अच्छे लोग जल्दी से दोस्त बन जाते हैं
2	हरेः पत्नी लक्ष्मीः	हरि की पत्नी लक्ष्मी है	लक्ष्मी हरि की पत्नी है
3	आपत्काले बुद्धेः परीक्षा भवति ।	आपत्काल में बुद्धि की परीक्षा होती है	आपातकालीन में बुद्धि का परीक्षण होता है
4	तत्र धेनूनां समूहः तिष्ठति	वहाँ गायों का समूह रहता है	गायों का समूह वहां रहता है
5	स्यात् स्वप्नः आकाशात् उच्चतरः सागरात् गभीरतरश्च ।	आकाश की तुलना में अधिक ऊँचा ड्रीम और महासागर से गहराई से	ख्वाब आकाश से ऊँचा और महासागर से गहरा होना चाहिए
6	अहम् गच्छामि	मैं जाता हूँ	मैं जाता हूँ
7	यदा आशानिवृतिः तदा शान्तिसमुद्भवः ।	शांति तब शुरू होती है जब अपेक्षा समाप्त होती है	जब अपेक्षा समाप्त होती है तब शांति शुरू होती है
8	यदा आशानिवृतिः तदा शान्तिसमुद्भवः ।	शांति तब शुरू होती है जब अपेक्षा समाप्त होती है	जब अपेक्षा समाप्त होती है तब शांति शुरू होती है
9	वचांसि हन्तुं प्रभवन्ति । वचनप्रयोगे भव अप्रमत्तः	शब्द मार सकते हैं सावधान रहें जब आप उन्हें प्रयोग कर रहे हों	शब्द मार सकते हैं उनका प्रयोग सावधानी से करे

into different modules or phases developed under Sanskrit Consortium Project funded by MIT using Anusaaraka[211]. There are 10 modules in the rule-based pipeline architecture of Sanskrit to Hindi translation (Available online at: http://sanskrit.uohyd.ac.in/scl). All of the modules provide an individual output as linguistic features to Neural based Encoder-Decoder to train the system more efficiently. The experiment has been performed on different Bi-RNN models by increasing the hidden layer and epochs, update, time, train and development probability along with GTX GPU versions as shown in Figure 5.4(a); and other details have been provided in the subsequent sections. The evaluation of the proposed system was made through automatic as well as human measures was shown. The automatic analysis of translated sentences was performed on the basis of parameters, such as BLEU, Precision, Recall, F-measure, WER, F-mean, Penalty and Meteor as manifested in Figure 5.4(b). Linguistically, the results have shown that the highest error rate of 4% in the proposed MT system is encountered in the sentences relating to the verb category, whereas other categories have less than 3% error rate. The proposed hybrid MTS has 61% accuracy, whereas the accuracy of the existing MTS 'SaHiT'[190]is only 27%. So, the results have show, that the proposed MTS performs much better than the existing MTS[190] for Sanskrit to Hindi Translation, and is a great contribution to the teaching-learning process.

5.3 Conclusion

The work aims at capturing and quantifying word-level errors and classifying them into a taxonomy. As a result, previous taxonomies have been expanded to adapt the errors transpired in morphological rich Indo-European languages. The error analysis is based on different linguistic levels and their sub-categories i.e., orthography, morphology, lexical, syntax, semantics and pragmatics. The proposed taxonomy has been compared with all the previous taxonomies based on various parameters such as i.e., categorization, mode,

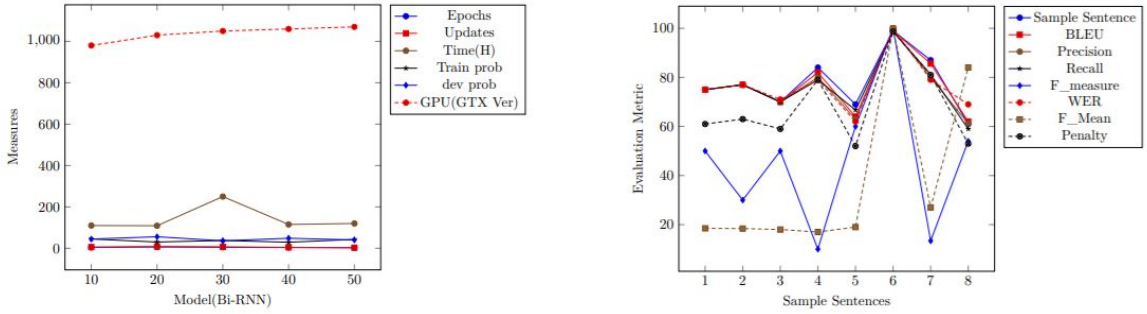


Figure 5.4: (a) Comparison of BI-RNN Model Across Different Parameters, (b) Evaluation of Sample Sanskrit Sentences Mentioned in Table 6.2 Across Different Measures

language dependency, MT approaches, translation pair, tool, domain, corpus and size of the corpus. Further, a case study of the proposed error taxonomy on Sanskrit-Hindi language has been undertaken. It involves a deep analysis of 24 different Sanskrit Corpora based on error location in different linguistic levels, i.e., orthography, morphology, lexical, syntax, semantic and pragmatics. The MT used for experimental work was developed in various phases. Firstly, the rule-based and neural-based model was developed. Later, by incorporating linguistic rich features of the rule-based system, a neural-model was trained to form it as a hybrid-based. Thus, the work presented here covers a long range of errors generated by neural-based, rule-based and hybrid-based MT. The work provides a possible solution to the different linguistic level errors. It enhances the readability, quality and productivity of the MTS. It contributes significantly towards the human error analysis process.

The proposed work has added new prospects in the teaching-learning paradigm. The MTS has been developed by using linguistic processing and NMT. This system can be accessed with a graphical interface with click of a button. It improves the teaching-learning process and substitutes the traditional methods which require 8-12 years of study. The work can be of great importance for the school as well as university student enrolled in distance education and certain age group people as it promotes self-learning and enables them to read and understand the philosophical texts of Gita, Ramayana and Mahabharta.

The following chapter concludes this entire thesis with directions for future work.

Chapter 6

Conclusions and Directions for Future Research

This chapter is the concluding part of the thesis and also proposes some suggestions towards which the present work can be further extended. Section 6.1 brings out the overall conclusions of the research work carried out in this thesis and in section 6.2 suggestions regarding the future research directions and possible extensions of the work presented in the thesis are made.

This thesis focuses on an extensive literature review which serves the purpose of providing resources required for modelling different techniques such as corpus, domains, toolkits, models, features and their evaluation measures. The work also carries the directions for future research in the field under study. The proposed system is an integration of linguistically rich and result oriented approaches called rule-based and neural-based approach are gaining significant attention nowadays. It consists of several heterogeneous modules. Deploying such an application on a stand-alone machine becomes a complex task. It has been observed that local server takes more time to respond and provides lesser accuracy. Therefore, offering MTS as a cloud service is a better proposition for increasing its performance in terms of accuracy and response time. Moreover, auto-tuning for neural-based MT is not possible at the local host due to memory issues, but it is possible on the cloud. The proposed CBSHH-MTS provides better throughput, rule matching probability and number of matching rules in comparison to the stand-alone systems.

6.1 Conclusions

In this thesis, an attempt has been made to provide an automatic translation of Sanskrit to Hindi with linguistic analysis. This system can be accessed with a graphical interface on the click of a button. It improves the teaching-learning process and substitutes the traditional methods which require 8-12 years of study. The work can be of great importance for the school as well as university student enrolled in distance education and certain age group people as it promotes self-learning and enables them to read and understand the philosophical texts of Gita, Ramayana and Mahabharta. The main contribution of this thesis is done in several phases as listed.

- Extensive literature survey of the work carried out in the area of machine translation was undertaken and presented for devising an effective MT technique.
- Proposed an SHH-MTS System merging linguistic tools output from the classical rule-based approach as features embedding matrices in NMT.
- To overcome the scarcity of data, firstly a parallel corpus of Bhagavad-Geeta was manually curated. Secondly, a synthetic parallel corpus is also curated.
- A web-interface was developed for better user-interface. The web-interface was deployed on AWS cloud with EC2 which eased time, knowledge and complications. It even becomes challenging for a common user to utilize such a complex application. It elevated the MTS performance by managing recurrent changes in terms of either corpus, domain, algorithm and rules.
- A case study of the proposed error taxonomy was also performed to validate the performance of the proposed machine translation system to achieve an effective translation from Sanskrit to Hindi.

6.2 Directions for Future Research

This section proposes some suggestions towards which the present work can be further extended in future in this area:

- Recent trend of deep learning in computer vision and speech recognition has inspired the MT field to develop deeper models. It reduces the model perplexity, increases the accuracy, and builds a better coverage model. Different neural architectures need to be explored, especially for the decoder.
- The technique for paraphrase creation needs to be explored from manual word substitutions to pivot technique of other translation systems.

- Usage of Ensemble techniques requires manual intervention. It is the merging of multiple alternative generative systems and combining outputs, of different systems.
- The system can be used to build E-reading and E-capsules for Sanskrit language learning and teaching. It is developed and trained at the elementary stage, and rare instances, derivational and pada formations can also be included in it. The system initial training data can be used for boot-strapping of automatic annotation of a complete book.
- The developed system can contribute to forming a Sanskrit Digital repository. It can be further utilized to create virtual courses with the same interface.
- The teaching-learning system can be extended to form a multilingual interface covering various other languages.

Bibliography

[1] Arturo Trujillo. *Translation engines: techniques for machine translation*. Springer Science & Business Media, 2012.

[2] Akshar Bharati, Vineet Chaitanya, Rajeev Sangal, and KV Ramakrishnamacharyulu. *Natural language processing: a Paninian perspective*. Prentice-Hall of India New Delhi, 1995.

[3] Gobinda G Chowdhury. Natural language processing. *Annual review of information science and technology*, 37(1):51–89, 2003.

[4] Mikel L Forcada, Mireia Ginestí-Rosell, Jacob Nordfalk, Jim O'Regan, Sergio Ortiz-Rojas, Juan Antonio Pérez-Ortiz, Felipe Sánchez-Martínez, Gema Ramírez-Sánchez, and Francis M Tyers. Apertium: a free/open-source platform for rule-based machine translation. *Machine translation*, 25(2):127–144, 2011.

[5] Grace Noone. Machine translation a transfer approach. *Computer Science, Linguistics and a Language (CSLL) Department, University of Dublin, Trinity College, Final Rep*, 2003.

[6] Shachi Dave, Jignashu Parikh, and Pushpak Bhattacharyya. Interlingua-based english-hindi machine translation and language divergence. *Machine Translation*, 16(4):251–304, 2001.

[7] Peter F Brown, Vincent J Della Pietra, Stephen A Della Pietra, and Robert L Mercer. The mathematics of statistical machine translation: Parameter estimation. *Computational linguistics*, 19(2):263–311, 1993.

[8] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[9] Harold Somers. Example-based machine translation. *Machine translation*, 14(2):113–157, 1999.

[10] Sergei Nirenburg. al.(1989) kbmt-89 project report. *Center for Machine Translation, Carnegie Mellon University, Pittsburg*, page 286, 1989.

[11] William John Hutchins and Harold L Somers. *An introduction to machine translation*, volume 362. Academic Press London, 1992.

[12] Mireia Farreús, Marta R Costa-jussà, and Maja Popović Morse. Study and correlation analysis of linguistic, perceptual, and automatic machine translation evaluations. *Journal of the american society for information science and technology*, 63(1):174–184, 2012.

[13] Philipp Koehn. *Statistical machine translation*. Cambridge University Press, 2009.

[14] Nal Kalchbrenner and Phil Blunsom. Recurrent continuous translation models. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1700–1709, 2013.

[15] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[16] Mikel L Forcada and Ramón P Ñeco. Recursive hetero-associative memories for translation. In *International Work-Conference on Artificial Neural Networks*, pages 453–462. Springer, 1997.

[17] Michel Simard, Nicola Ueffing, Pierre Isabelle, and Roland Kuhn. Rule-based translation with statistical phrase-based post-editing. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 203–206. Association for Computational Linguistics, 2007.

[18] Declan Groves and Andy Way. Hybrid data-driven models of machine translation. *Machine Translation*, 19(3-4):301–323, 2005.

[19] Goverment of India. Census-2018,"language-census of india, states and union territories", 2018. Accessed on: 11-03-2019.

[20] Meera Baindur. *Nature in Indian philosophy and cultural traditions*. Springer, 2015.

[21] Girish N Jha, Sudhir K Mishra, and R Chandrashekar. Developing a sanskrit analysis system for machine translation. In *Proc. National Seminar on Translation Today: state and issues, Dept. of Linguistics, University of Kerala, Trivandrum*, pages 23–25, 2005.

[22] Gérard Huet. Formal structure of sanskrit text: Requirements analysis for a mechanical sanskrit processor. In *Sanskrit Computational Linguistics*, pages 162–199. Springer, 2009.

[23] Akshar Bharati, Vineet Chaitanya, and Rajeev Sangal. Paninian framework and its application toanusaraka. *Sadhana*, 19(1):113–127, 1994.

[24] Akshar Bharati and Amba Kulkarni. Sanskrit and computational linguistics. In *First International Sanskrit Computational Symposium. Department of Sanskrit Studies, University of Hyderabad*, 2007.

[25] R Raman Nair and L Sulochana Devi. *Sanskrit Informatics: Informatics for Sanskrit studies and research*. Centre for Informatics Research an dDevelopment, 2011.

[26] Preeti Shukla, Devanand Shukl, and Amba Kulkarni. Vibhakti divergence between sanskrit and hindi. In *International Sanskrit Computational Linguistics Symposium*, pages 198–208. Springer, 2010.

[27] Armando Fox, Rean Griffith, Anthony Joseph, Randy Katz, Andrew Konwinski, Gunho Lee, David Patterson, Ariel Rabkin, and Ion Stoica. Above the clouds: A berkeley view of cloud computing. *Dept. Electrical Eng. and Comput. Sciences, University of California, Berkeley, Rep. UCB/EECS*, 28(13):1–25, 2009. http://home.cse.ust.hk/ weiwa/teaching/Fall15 - COMP6611B/reading_list/AboveTheClouds.pdf.

[28] George Pallis. Cloud computing: the new frontier of internet computing. *IEEE Internet Computing*, 14(5):70–73, 2010.

[29] K Thirupathi Rao, P Sai Kiran, and L Siva Shanker Reddy. Energy efficiency in datacenters through virtualization: A case study. *Global Journal of Computer Science and Technology*, 10(3), 2010.

[30] Anton Beloglazov and Rajkumar Buyya. Energy efficient resource management in virtualized cloud data centers. In *Proceedings of the 2010 10th IEEE/ACM international conference on cluster, cloud and grid computing*, pages 826–831. IEEE Computer Society, 2010. https://dl.acm.org/purchase.cfm?id=1845139 &CFID=828049556&CFTOKEN=22335400.

[31] Luis M Vaquero, Luis Rodero-Merino, Juan Caceres, and Maik Lindner. A break in the clouds: towards a cloud definition. *ACM SIGCOMM Computer Communication Review*, 39(1):50–55, 2008.

[32] Mladen A Vouk. Cloud computing–issues, research and implementations. *CIT. Journal of Computing and Information Technology*, 16(4):235–246, 2008.

[33] Saurabh Kumar Garg, Chee Shin Yeo, Arun Anandasivam, and Rajkumar Buyya. Environment-conscious scheduling of hpc applications on distributed cloud-oriented data centers. *Journal of Parallel and Distributed Computing*, 71(6):732–749, 2011.

[34] Robert W Lucky. Cloud computing [reflections]. *IEEE Spectrum*, 46(5):27–45, 2009.

[35] Marios D Dikaiakos, Dimitrios Katsaros, Pankaj Mehra, George Pallis, and Athena Vakali. Cloud computing: Distributed internet computing for it and scientific research. *Internet Computing, IEEE*, 13(5):10–13, 2009.

[36] Ajith Singh and M Hemalatha. Cloud computing for academic environment. *International Journal of Information and Communication Technology Research*, 2(2):97–101, 2012.

[37] Ilango Sriram and Ali Khajeh-Hosseini. Research agenda in cloud technologies. *arXiv preprint arXiv:1001.3259*, 2010. https://arxiv.org/ftp/arxiv/papers/1001/1001.3259.pdf.

[38] Gabor Kecskemeti, Mark Gergely, Adam Visegradi, Zsolt Nemeth, Jozsef Kovacs, and Péter Kacsuk. One click cloud orchestrator: Bringing complex applications effortlessly to the clouds. In *European Conference on Parallel Processing*, pages 38–49. Springer, 2014. http://link.springer.com/chapter/10.1007/978-3-319-14313-2_4.

[39] Qi Zhang, Lu Cheng, and Raouf Boutaba. Cloud computing: state-of-the-art and research challenges. *Journal of internet services and applications*, 1(1):7–18, 2010.

[40] Sunil Kumar Chowdhary, Ajit Yadav, and Naveen Garg. Cloud computing: Future prospect for e-health. In *Electronics Computer Technology (ICECT), 2011 3rd International Conference on*, volume 3, pages 297–299. IEEE, 2011.

[41] Fei Teng. *Management of Data and Scheduling of Tasks on Architecture Distributees*. PhD thesis, Ph. D. thesis. École Centrale: A University Institution, Paris, 2011.

[42] Bhaskar Prasad Rimal, Eunmi Choi, and Ian Lumb. A taxonomy and survey of cloud computing systems. In *2009 Fifth International Joint Conference on INC, IMS and IDC*, pages 44–51. Ieee, 2009. http://s3.amazonaws.com/academia.edu.documents/34625815/A_Taxonomy_and_Survey_of_Cloud_Computing_System.pdf.

[43] Lizhe Wang, Gregor Von Laszewski, Andrew Younge, Xi He, Marcel Kunze, Jie Tao, and Cheng Fu. Cloud computing: a perspective study. *New Generation Computing*, 28(2):137–146, 2010. http://dx.doi.org/10.1007/s00354-008-0081-5.

[44] David Hilley. Cloud computing: A taxonomy of platform and infrastructure-level offerings. *Georgia Institute of Technology, Tech. Rep*, page 1—38, 2009. http://www.cercs.gatech.edu/tech-reports/tr2009/ git-cercs-09-13.pdf.

[45] Armando Fox, Rean Griffith, Anthony Joseph, Randy Katz, Andrew Konwinski, Gunho Lee, David Patterson, Ariel Rabkin, and Ion Stoica. Above the clouds: A berkeley view of cloud computing. *Dept. Electrical Eng. and Comput. Sciences, University of California, Berkeley, Rep. UCB/EECS*, 28(13):2009, 2009.

[46] Ian Foster, Yong Zhao, Ioan Raicu, and Shiyong Lu. Cloud computing and grid computing 360-degree compared. In *Grid Computing Environments Workshop*, pages 1–10. IEEE, 2008.

[47] Ishfaq Ahmad and Sanjay Ranka. *Handbook of Energy-Aware and Green Computing-Two Volume Set*. CRC Press, 2016.

[48] Bhaskar Prasad Rimal, Eunmi Choi, and Ian Lumb. A taxonomy and survey of cloud computing systems. In *INC, IMS and IDC, 2009. NCM'09. Fifth International Joint Conference on*, pages 44–51. Ieee, 2009.

[49] Pawan Kumar, Rashid Ahmad, Banshi Dhar Chaudhary, and Rajeev Sangal. Machine translation system as virtual appliance: For scalable service deployment on cloud. In *IEEE Seventh International Symposium on Service-Oriented System Engineering*, pages 304–308. IEEE, 2013.

[50] Weidong Shi, Yang Lu, Zhu Li, and Jonathan Engelsma. Sharc: A scalable 3d graphics virtual appliance delivery framework in cloud. *Journal of Network and Computer Applications*, 34(4):1078–1087, 2011.

[51] Christian Schwartze, Sven Kralisch, and Wolfgang-Albert Flügel. Geospatial virtual appliances using open source software. In *International Symposium on Environmental Software Systems*, pages 154–160. Springer, 2011.

[52] Michael Abd-El-Malek, Matthew Wachs, James Cipar, Karan Sanghi, Gregory R Ganger, Garth A Gibson, and Michael K Reiter. File system virtual appliances: Portable file system implementations. *ACM Transactions on Storage (TOS)*, 8(3):9, 2012.

[53] Oren Laadan, Jason Nieh, and Nicolas Viennot. Teaching operating systems using virtual appliances and distributed version control. In *Proceedings of the 41st ACM Technical Symposium on Computer Science Education*, pages 480–484. ACM, 2010.

[54] Fumio Machida, Masahiro Kawato, and Yoshiharu Maeno. Renovating legacy distributed systems using virtual appliance with dependency graph. In *2010 International Conference on Network and Service Management*, pages 17–24. IEEE, 2010.

[55] W John Hutchins. Machine translation over fifty years. *Histoire epistémologie langage*, 23(1):7–31, 2001.

[56] Bonnie J Dorr, Eduard H Hovy, and Lori S Levin. Machine translation: Interlingual methods. *Natural Language Processing and Machine Translation Encyclopedia of Language and Linguistics, 2nd ed. (ELL2).*, 2004.

[57] B Kavirajan, M Anand Kumar, KP Soman, S Rajendran, and S Vaithehi. Improving the rule based machine translation system using sentence simplification (english to tamil). In *2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pages 957–963. IEEE, 2017.

[58] Manish Rana and Mohammad Atique. Use of fuzzy tool for example based machine translation. *Procedia Computer Science*, 79:199–206, 2016.

[59] Hemant Darbari, Ajai Kumar, Aparupa Dasgupta, Sudhir K Mishra, Pune C-DAC, and Pune C-DAC. Complexity of language in rajya sabha domain and mantra approach.

[60] GV Garje, GK Kharate, and Harshad Kulkarni. Transmuter: an approach to rule-based english to marathi machine translation. *International Journal of Computer Applications*, 98(21), 2014.

[61] CCS Basavaraddi and HL Shashirekha. A typical mt system for english to kannada. *Int J Sci Eng Res*, 5(4), 2014.

[62] Abhay Adapanawar, Anita Garje, Paurnima Thakare, Prajakta Gundawar, and Priyanka Kulkarni. Rule based english to marathi translation of assertive sentence. *International Journal Of Scientific & Engineering Research*, 4(5), 2013.

[63] Devika Pishartoy and Sayli Wandkar Priya. Extending capabilities of english to marathi machine translator. *I JCSI International Journal of Computer Science Issues*, 9(3), 2012.

[64] Latha R Nair, David Peter, and Renjith P Ravindran. Design and development of a malayalam to english translator-a transfer based approach. *International Journal of Computational Linguistics (IJCL)*, 3(1):1–11, 2012.

[65] Kamaljeet Kaur Batra and GS Lehal. Rule based machine translation of noun phrases from punjabi to english. *International Journal of Computer Science Issues (IJCSI)*, 7(5):409, 2010.

[66] Remya Rajan, Remya Sivan, Remya Ravindran, and KP Soman. Rule based machine translation from english to malayalam. In *Advances in Computing, Control, & Telecommunication Technologies, 2009. ACT'09. International Conference on*, pages 439–441. IEEE, 2009.

[67] R Mahesh K Sinha and K Mahesh. Developing english-urdu machine translation via hindi. In *Third Workshop on Computational Approaches to Arabic-Script-based Languages*. Citeseer, 2009.

[68] RMK Sinha and A Jain. Anglahindi: an english to hindi machine-aided translation system. *MT Summit IX, New Orleans, USA*, pages 494–497, 2003.

[69] M Anand Kumar, B Premjith, S Shivkaran, B Kavirajan, A Rajendran, and KP Soman. Overview of the shared task on machine translation in indian languages (mtil-2017). *J. Intell. Syst*, 2017.

[70] Chandranath Adak. A bilingual machine translation system: English & bengali. In *Automation, Control, Energy and Systems (ACES), 2014 First International Conference on*, pages 1–4. IEEE, 2014.

[71] Preeti Dubey. Need for hindi-dogri machine translation system. In *Computing for Sustainable Global Development (INDIACom), 2014 International Conference on*, pages 136–140. IEEE, 2014.

[72] Warren Weaver. Translation. *Machine translation of languages*, 14:15–23, 1955.

[73] Stephan Vogel, Hermann Ney, and Christoph Tillmann. Hmm-based word alignment in statistical translation. In *Proceedings of the 16th conference on Computational linguistics-Volume 2*, pages 836–841. Association for Computational Linguistics, 1996.

[74] Franz Josef Och and Hermann Ney. A systematic comparison of various statistical alignment models. *Computational linguistics*, 29(1):19–51, 2003.

[75] Philipp Koehn, Franz Josef Och, and Daniel Marcu. Statistical phrase-based translation. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 48–54. Association for Computational Linguistics, 2003.

[76] Kenji Yamada and Kevin Knight. A syntax-based statistical translation model. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics*, 2001.

[77] Eugene Charniak, Kevin Knight, and Kenji Yamada. Syntax-based language models for statistical machine translation. In *Proceedings of MT Summit IX*, pages 40–46, 2003.

[78] David Chiang. A hierarchical phrase-based model for statistical machine translation. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 263–270. Association for Computational Linguistics, 2005.

[79] Basim Shahid Baqui Subalalitha, Aarthi Venkataraman. Statistical machine translation system from english-hindi. *International Journal of Pure and Applied Mathematics*, 118(20):1649–1655, 2018.

[80] Shishpal Jindal, Vishal Goyal, and Jaskarn Singh Bhullar. English to punjabi statistical machine translation using moses (corpus based). *Journal of Statistics and Management Systems*, 21(4):553–560, 2018.

[81] Raj Nath Patel, Prakash B Pimpale, and M Sasikumar. Machine translation in indian languages: Challenges and resolution. *Journal of Intelligent Systems*, 2018.

[82] Nadeem Jadoon Khan, Waqas Anwar, and Nadir Durrani. Machine translation approaches and survey for indian languages. *arXiv preprint arXiv:1701.04290*, 2017.

[83] Raj Nath Patel, Prakash B Pimpale, et al. Statistical machine translation for indian languages: Mission hindi. *arXiv preprint arXiv:1610.07418*, 2016.

[84] Raj Nath Patel and Prakash B Pimpale. Statistical machine translation for indian languages: Mission hindi2. *arXiv:1610.07418v1*, 2016.

[85] Pranjal Das and Kalyanee K Baruah. Assamese to english statistical machine translation integrated with a transliteration module. *International Journal of Computer Applications*, 100(5), 2014.

[86] Aasim Ali, Shahid Siddiq, and Muhammad Kamran Malik. Development of parallel corpus and english to urdu statistical machine translation. *Int. J. of Engineering & Technology IJET-IJENS*, 10:31–33, 2010.

[87] Nadeem Khan, Muhammad Waqas Anwar, Usama Ijaz Bajwa, and Nadir Durrani. English to urdu hierarchical phrase-based statistical machine translation. In *Proceedings of the 4th Workshop on South and Southeast Asian Natural Language Processing*, pages 72–76, 2013.

[88] Aasim Ali, Arshad Hussain, and Muhammad Kamran Malik. Model for english-urdu statistical machine translation. *World Applied Sciences*, 24:1362–1367, 2013.

[89] Pankaj Kumar and V Kumar. Statistical machine translation based punjabi to english transliteration system for proper nouns. *International Journal of Application or Innovation in Engineering & Management*, 2(8):318–321, 2013.

[90] Md Musfique Anwar, Mohammad Zabed Anwar, and Md Al-Amin Bhuiyan. Syntax analysis and machine translation of bangla sentences. *International Journal of Computer Science and Network Security*, 9(8):317–326, 2009.

[91] Raghavendra Udupa and Tanveer A Faruquie. An english-hindi statistical machine translation system. In *International Conference on Natural Language Processing*, pages 254–262. Springer, 2004.

[92] Colin Cherry. Cohesive phrase-based decoding for statistical machine translation. *Proceedings of ACL-08: HLT*, pages 72–80, 2008.

[93] Saurav Jha, Akhilesh Sudhakar, and Anil Kumar Singh. Neural machine translation based word transduction mechanisms for low-resource languages. *arXiv preprint arXiv:1811.08816*, 2018.

[94] Himanshu Choudhary, Aditya Kumar Pathak, Rajiv Ratan Saha, and Ponnurangam Kumaraguru. Neural machine translation for english-tamil. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 770–775, 2018.

[95] Amarnath Pathak and Partha Pakray. Neural machine translation for indian languages. *Journal of Intelligent Systems*, 2017.

[96] Sree Harsha Ramesh and Krishna Prasad Sankaranarayanan. Neural machine translation for low resource languages using bilingual lexicon induced from comparable corpora. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Student Research Workshop*, pages 112–119, 2018.

[97] Shivkaran Singh, M Anand Kumar, and KP Soman. Attention based english to punjabi neural machine translation. *Journal of Intelligent & Fuzzy Systems*, 34(3):1551–1559, 2018.

[98] Jigar Mistry, Ajay Anand Verma, and Pushpak Bhattacharyya. Literature survey: Study of neural machine translation.

[99] Karthik Revanuru, Kaushik Turlapaty, and Shrisha Rao. Neural machine translation of indian languages. In *Proceedings of the 10th Annual ACM India Compute Conference on ZZZ*, pages 11–20. ACM, 2017.

[100] Pasindu Tennage, Prabath Sandaruwan, Malith Thilakarathne, Achini Herath, Surangika Ranathunga, Sanath Jayasena, and Gihan Dias. Neural machine translation for sinhala and tamil languages. In *2017 International Conference on Asian Language Processing (IALP)*, pages 189–192. IEEE, 2017.

[101] Ruchit Agrawal and Dipti Misra Sharma. Building an effective mt system for english-hindi using rnn's. *International Journal of Artificial Intelligence and Applications (IJAIA)*, 8(5), 2017.

[102] Ayan Das, Pranay Yerra, Ken Kumar, and Sudeshna Sarkar. A study of attention-based neural machine translation model on indian languages. In *Proceedings of the 6th Workshop on South and Southeast Asian Natural Language Processing (WSSANLP2016)*, pages 163–172, 2016.

[103] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014.

[104] Minh-Thang Luong, Quoc V Le, Ilya Sutskever, Oriol Vinyals, and Lukasz Kaiser. Multi-task sequence to sequence learning. *arXiv preprint arXiv:1511.06114*, 2015.

[105] Sébastien Jean, Orhan Firat, Kyunghyun Cho, Roland Memisevic, and Yoshua Bengio. Montreal neural machine translation systems for wmt'15. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 134–140, 2015.

[106] Minh-Thang Luong, Ilya Sutskever, Quoc V Le, Oriol Vinyals, and Wojciech Zaremba. Addressing the rare word problem in neural machine translation. *arXiv preprint arXiv:1410.8206*, 2014.

[107] Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*, 2014.

[108] Jan K Chorowski, Dzmitry Bahdanau, Dmitriy Serdyuk, Kyunghyun Cho, and Yoshua Bengio. Attention-based models for speech recognition. In *Advances in neural information processing systems*, pages 577–585, 2015.

[109] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[110] Minh-Thang Luong, Hieu Pham, and Christopher D Manning. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*, 2015.

[111] Fandong Meng, Zhengdong Lu, Mingxuan Wang, Hang Li, Wenbin Jiang, and Qun Liu. Encoding source language with convolutional neural network for machine translation. *arXiv preprint arXiv:1503.01838*, 2015.

[112] Łukasz Kaiser and Samy Bengio. Can active memory replace attention? In *Advances in Neural Information Processing Systems*, pages 3781–3789, 2016.

[113] Nal Kalchbrenner, Lasse Espeholt, Karen Simonyan, Aaron van den Oord, Alex Graves, and Koray Kavukcuoglu. Neural machine translation in linear time. *arXiv preprint arXiv:1610.10099*, 2016.

[114] Lukasz Kaiser, Aidan N Gomez, and Francois Chollet. Depthwise separable convolutions for neural machine translation. *arXiv preprint arXiv:1706.03059*, 2017.

[115] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 2016.

[116] Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. Convolutional sequence to sequence learning. corr abs/1705.03122, 2017.

[117] Nizar Habash, Bonnie Dorr, and Christof Monz. Symbolic-to-statistical hybridization: extending generation-heavy machine translation. *Machine Translation*, 23(1):23–63, 2009.

[118] Felipe Sánchez-Martínez, Mikel L Forcada, Andy Way, et al. Hybrid rule-based-example-based mt: feeding apertium with sub-sentential translation units. *Proceedings of the 3rd Workshop on Example Based Machine Translation*, pages 11–18, 2009.

[119] Alexandra Antonova and Alexey Misyurev. Improving the precision of automatically constructed human-oriented translation dictionaries. In *Proceedings of the 3rd Workshop on Hybrid Approaches to Machine Translation (HyTra)*, pages 58–66, 2014.

[120] Anne Göhring. Building a spanish-german dictionary for hybrid mt. In *Proceedings of the 3rd Workshop on Hybrid Approaches to Machine Translation (HyTra)*, pages 30–35, 2014.

[121] Felipe Sánchez-Martínez and Mikel L Forcada. Inferring shallow-transfer machine translation rules from small parallel corpora. *Journal of Artificial Intelligence Research*, 34:605–635, 2009.

[122] Francis M Tyers, Felipe Sánchez-Martínez, Mikel L Forcada, et al. Flexible finite-state lexical selection for rule-based machine translation. *Proceedings of the 16th EAMT Conference,Trento, Italy*, 2012.

[123] Alex Rudnick and Michael Gasser. Lexical selection for hybrid mt with sequence labeling. In *Proceedings of the Second Workshop on Hybrid Approaches to Translation*, pages 102–108, 2013.

[124] Marta Ruiz Costa-Jussà and Jordi Centelles. Description of the chinese-to-spanish rule-based machine translation system developed with a hybrid combination of human annotation and statistical techniques. *ACM transactions on asian language information processing*, 15(1):1–13, 2015.

[125] Christian Federmann and Sabine Hunsicker. Stochastic parse tree selection for an existing rbmt system. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 351–357. Association for Computational Linguistics, 2011.

[126] Catherine Dove, Olga Loskutova, and Ruben de la Fuente. What's your pick: Rbmt, smt or hybrid. In *Proceedings of the tenth conference of the Association for Machine Translation in the Americas (AMTA 2012). San Diego, CA*, 2012.

[127] Sabine Hunsicker, Chen Yu, and Christian Federmann. Machine learning for hybrid machine translation. In *Proceedings of the seventh workshop on statistical machine translation*, pages 312–316. Association for Computational Linguistics, 2012.

[128] Gorka Labaka, Cristina España-Bonet, Lluís Màrquez, and Kepa Sarasola. A hybrid machine translation architecture guided by syntax. *Machine translation*, 28(2):91–125, 2014.

[129] J Crego. Systran rbmt engine: hybridization experiments. In *3rd Workshop on Hybrid Approaches to Machine Translation (HyTra), Gothenburg, Sweden*, 2014.

[130] Kurt Eberle. Hybrid strategies for better products and shorter time-to-market. In *Proceedings of the 3rd Workshop on Hybrid Approaches to Machine Translation (HyTra)*, page 97, 2014.

[131] A-L Lagarda, Vicente Alabau, Francisco Casacuberta, Roberto Silva, and Enrique Díaz-de Liaño. Statistical post-editing of a rule-based machine translation system. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Short Papers*, pages 217–220. Association for Computational Linguistics, 2009.

[132] Hirokazu Suzuki. Automatic post-editing based on smt and its selective application by sentence-level automatic quality evaluation. *Language*, 1:59–429, 2011.

[133] Hanna Béchara, Raphaël Rubino, Yifan He, Yanjun Ma, and Josef van Genabith. An evaluation of statistical post-editing systems applied to rbmt and smt systems. *Proceedings of COLING 2012*, pages 215–230, 2012.

[134] Fei Xia and Michael McCord. Improving a statistical mt system with automatically learned rewrite patterns. In *Proceedings of the 20th international conference on Computational Linguistics*, page 508. Association for Computational Linguistics, 2004.

[135] Michael Collins, Philipp Koehn, and Ivona Kučerová. Clause restructuring for statistical machine translation. In *Proceedings of the 43rd annual meeting on association for computational linguistics*, pages 531–540. Association for Computational Linguistics, 2005.

[136] Chao Wang, Michael Collins, and Philipp Koehn. Chinese syntactic reordering for statistical machine translation. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, 2007.

[137] Raj Nath Patel, Rohit Gupta, Prakash B Pimpale, et al. Reordering rules for english-hindi smt. *arXiv preprint arXiv:1610.07420*, 2016.

[138] Mireia Farrús, Marta R Costa-Jussa, José B Mariño, Marc Poch, Adolfo Hernández, Carlos Henríquez, and José AR Fonollosa. Overcoming statistical machine translation limitations: error analysis and proposed solutions for the catalan–spanish language pair. *Language resources and evaluation*, 45(2):181–208, 2011.

[139] Lluís Formiga Fanals, Adolfo Hernández Huerta, José Bernardo Mariño Acebal, and Enrique Monte Moreno. Improving english to spanish out-of-domain translations by morphology generalization and generation. In *Proceedings of the Monolingual Machine Translation-2012 Workshop*, pages 6–16, 2012.

[140] Michael Carl, Cathrine Pease, Leonid L Iomdin, and Oliver Streiter. Towards a dynamic linkage of example-based and rule-based machine translation. *Machine Translation*, 15(3):223–257, 2000.

[141] Wu Hua and Wang Haifeng. Improving statistical word alignment with a rule-based machine translation system. In *Proceedings of the 20th international conference on Computational Linguistics*, page 29. Association for Computational Linguistics, 2004.

[142] Hideo Okuma, Hirofumi Yamamoto, and Eiichiro Sumita. Introducing a translation dictionary into phrase-based smt. *IEICE transactions on information and systems*, 91(7):2051–2057, 2008.

[143] Andreas Eisele, Christian Federmann, Hervé Saint-Amand, Michael Jellinghaus, Teresa Herrmann, and Yu Chen. Using moses to integrate multiple rule-based machine translation engines into a hybrid system. In *Proceedings of the Third Workshop on Statistical Machine Translation*, pages 179–182, 2008.

[144] Víctor M Sánchez-Cartagena, Felipe Sánchez-Martínez, Juan Antonio Pérez-Ortiz, et al. Integrating shallow-transfer rules into phrase-based statistical machine translation. In *Machine Translation Summit*, 2011.

[145] Yu Chen and Andreas Eisele. Integrating a rule-based with a hierarchical translation system. In *LREC*, 2010.

[146] Arafat Ahsan, Prasanth Kolachina, Sudheer Kolachina, Dipti Misra Sharma, and Rajeev Sangal. Coupling statistical machine translation with rule-based transfer and generation. In *Proceedings of the 9th Conference of the Association for Machine Translation in the Americas*, 2010.

[147] Ramona Enache, Cristina España Bonet, Aarne Ranta, and Lluís Màrquez Villodre. A hybrid system for patent translation. In *Proceedings of the 16th Annual Conference of the European Association for Machine Translation: EAMT 2012: Trento, Italy, May 28th-30th 2012*, pages 269–278, 2012.

[148] Franz Josef Och and Hermann Ney. The alignment template approach to statistical machine translation. *Computational linguistics*, 30(4):417–449, 2004.

[149] Kun Wang, Chengqing Zong, and Keh-Yih Su. Integrating translation memory into phrase-based machine translation during decoding. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 11–21, 2013.

[150] Jaime G Carbonell, Steve Klein, David Miller, Mike Steinbaum, Tomer Grassiany, and Jochen Frey. Context-based machine translation. *7th Conference of the Association for Machine Translation in the Americas*, 2006.

[151] Vincent Vandeghinste, Ineke Schuurman, Michael Carl, Stella Markantonatou, and Toni Badia. Metis-ii: Machine translation for low resource languages. In *LREC*, pages 1284–1289, 2006.

[152] Marta Ruiz Costa-Jussà and José Adrián Rodríguez Fonollosa. Using linear interpolation and weighted reordering hypotheses in the moses system. In *Seventh Conference on International Language Resources and Evaluation*, pages 1712–1718, 2011.

[153] George Tambouratzis, Sokratis Sofianopoulos, and Marina Vassiliou. Language-independent hybrid mt with presemt. In *Proceedings of the Second Workshop on Hybrid Approaches to Translation*, pages 123–130, 2013.

[154] Omkar Dhariya, Shrikant Malviya, and Uma Shanker Tiwary. A hybrid approach for hindi-english machine translation. In *2017 International Conference on Information Networking (ICOIN)*, pages 389–394. IEEE, 2017.

[155] Pramod Salunkhe, Aniket D Kadam, Shashank Joshi, Shuhas Patil, Devendrasingh Thakore, and Shrikant Jadhav. Hybrid machine translation for english to marathi: A research evaluation in machine translation:(hybrid translator). In *2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT)*, pages 924–931. IEEE, 2016.

[156] B Nithya and Shibily Joseph. A hybrid approach to english to malayalam machine translation. *International Journal of Computer Applications*, 81(8), 2013.

[157] Harjinder Kaur and Dr Vijay Laxmi. A web based english to punjabi mt system for news headlines. *International Journal of Advanced Research in Computer Science and Software Engineering*, 3(6):1092–1094, 2013.

[158] ML Dhore, SK Dixit, and JB Karande. Web page interface localisation in devanagari for commercial interactive applications by enhancing basic functionality of apache server. *International Journal of Computer Applications*, 18(4):6–10, 2011.

[159] Sanjay Chatterji, Praveen Sonare, Sudeshna Sarkar, and Anupam Basu. Lattice based lexical transfer in bengali hindi machine translation framework. In *Proceedings of ICON-2011: 9th International Conference on Natural Language Processing*, 2011.

[160] Shahnawaz and RB Mishra. An english to urdu translation model based on cbr, ann and translation rules. *International Journal of Advanced Intelligence Paradigms*, 7(1):1–23, 2015.

[161] Sanjay Chatterji, Devshri Roy, Sudeshna Sarkar, and Anupam Basu. A hybrid approach for bengali to hindi machine translation. In *Proceedings of ICON-2009: 7th International Conference on Natural Language Processing*, pages 81–91, 2009.

[162] Bonnie J Dorr. Machine translation divergences: A formal description and proposed solution. *Computational Linguistics*, 20(4):597–633, 1994.

[163] Nizar Habash and Bonnie Dorr. Handling translation divergences: Combining statistical and symbolic techniques in generation-heavy machine translation. In *Conference of the Association for Machine Translation in the Americas*, pages 84–93. Springer, 2002.

[164] Pawan Goyal and R Mahesh K Sinha. Translation divergence in english-sanskrit-hindi language pairs. In *International Sanskrit Computational Linguistics Symposium*, pages 134–143. Springer, 2009.

[165] Vimal Mishra and RB Mishra. Study of example based english to sanskrit machine translation. *Polibits*, (37):43–54, 2008.

[166] Amba Kulkarni and Monali Das. Discourse analysis of sanskrit texts. In *Proceedings of the Workshop on Advances in Discourse Analysis and its Computational Aspects*, pages 1–16, 2012.

[167] Minh-Thang Luong and Christopher D Manning. Stanford neural machine translation systems for spoken language domains. In *Proceedings of the International Workshop on Spoken Language Translation*, pages 76–79, 2015.

[168] M Amin Farajian, Marco Turchi, Matteo Negri, Nicola Bertoldi, and Marcello Federico. Neural vs. phrase-based machine translation in a multi-domain scenario. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, volume 2, pages 280–284, 2017.

[169] Marco Turchi, Tijl De Bie, and Nello Cristianini. Learning performance of a machine translation system: a statistical and computational analysis. In *Proceedings of the Third Workshop on Statistical Machine Translation*, pages 35–43. Association for Computational Linguistics, 2008.

[170] Ann Irvine and Chris Callison-Burch. Combining bilingual and comparable corpora for low resource machine translation. In *Proceedings of the eighth workshop on statistical machine translation*, pages 262–270, 2013.

[171] Boxing Chen, Roland Kuhn, George Foster, Colin Cherry, and Fei Huang. Bilingual methods for adaptive training data selection for machine translation. In *Proc. of AMTA*, pages 93–103, 2016.

[172] Philip Arthur, Graham Neubig, and Satoshi Nakamura. Incorporating discrete translation lexicons into neural machine translation. *arXiv preprint arXiv:1606.02006*, 2016.

[173] Zhaopeng Tu, Zhengdong Lu, Yang Liu, Xiaohua Liu, and Hang Li. Modeling coverage for neural machine translation. *arXiv preprint arXiv:1601.04811*, 2016.

[174] Wenhu Chen, Evgeny Matusov, Shahram Khadivi, and Jan-Thorsten Peter. Guided alignment training for topic-aware neural machine translation. *arXiv preprint arXiv:1607.01628*, 2016.

[175] Lemao Liu, Masao Utiyama, Andrew Finch, and Eiichiro Sumita. Neural machine translation with supervised attention. *arXiv preprint arXiv:1609.04186*, 2016.

[176] Yang Liu, Qun Liu, and Shouxun Lin. Tree-to-string alignment template for statistical machine translation. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 609–616. Association for Computational Linguistics, 2006.

[177] Franz Josef Och and Hermann Ney. A comparison of alignment models for statistical machine translation. In *COLING 2000 Volume 2: The 18th International Conference on Computational Linguistics*, volume 2, 2000.

[178] Liang Tian, Fai Wong, and Sam Chao. Word alignment using giza++ on windows. *Machine Translation*, 2011.

[179] Yonggang Deng and William Byrne. Mttk: An alignment toolkit for statistical machine translation. In *Proceedings of the 2006 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology: companion volume: demonstrations*, pages 265–268. Association for Computational Linguistics, 2006.

[180] Daniel Ortiz-Martínez, Ismael García-Varea, and Francisco Casacuberta. Thot: a toolkit to train phrase-based statistical translation models. In *In Tenth Machine Translation*. Citeseer, 2005.

[181] Philipp Koehn and Rebecca Knowles. Six challenges for neural machine translation. *arXiv preprint arXiv:1706.03872*, 2017.

[182] MD Okpor. Machine translation approaches: issues and challenges. *International Journal of Computer Science Issues (IJCSI)*, 11(5):159, 2014.

[183] Philipp Koehn, Amittai Axelrod, Alexandra Birch Mayne, Chris Callison-Burch, Miles Osborne, David Talbot, and Michael White. Edinburgh system description for the 2005 nist mt evaluation. In *Proceedings of Machine Translation Evaluation Workshop 2005*, 2005.

[184] Prasad Pingali and Vasudeva Varma. Hindi and telugu to english cross language information retrieval at clef 2006. In *CLEF (Working Notes)*, 2006.

[185] ML Dhore and SX Dixit. English to devnagari translation for ui labels of commercial web based interactive applications. *International Journal of Computer Applications*, 35(10):0975–8887, 2011.

[186] Amruta Godase and Sharvari Govilkar. A novel approach for rule based translation of english to marathi. *Advanced Computational Intelligence: An International Journal*, 2(4), 2015.

[187] Seema Baghla Savita Singla. Hybrid approach for english to punjabi translation system for news paper headlines in a specific domain. *International Journal of Engineering Research and Technology*, 2, 2013.

[188] Ruchika A Sinhal and Kapil O Gupta. A pure ebmt approach for english to hindi sentence translation system. *International Journal of Modern Education and Computer Science*, 6(7):1, 2014.

[189] ES Anju and KV Manoj Kumar. Malayalam to english machine translation: An ebmt system. *IOSR Journal of Engineering (IOSRJEN)*, 4(01):18–23, 2014.

[190] Rajneesh Kumar Pandey and Girish Nath Jha. Error analysis of sahit-a statistical sanskrit-hindi translator. *Procedia Computer Science*, 96:495–501, 2016.

[191] Sarita G Rathod. Machine translation of natural language using different approaches: Etsts (english to sanskrit translator and synthesizer). *International Journal of Computer Applications*, 102(15), 2014.

[192] Pragya Shukla and Akanksha Shukla. English speech to sanskrit speech (esss) using rule based translation. *International Journal of Computer Applications*, 92(10), 2014.

[193] Promila Bahadur, AK Jain, and DS Chauhan. Etrans-a complete framework for english to sanskrit machine translation. In *International Journal of Advanced Computer Science and Applications (IJACSA) from International Conference and workshop on Emerging Trends in Technology*. Citeseer, 2012.

[194] V Jayan, R Sunil, G Sulochana Kurambath, and R Ravindra Kumar. Divergence patterns in machine translation between malayalam and english. In *Proceedings of the International Conference on Advances in Computing, Communications and Informatics*, pages 788–794. ACM, 2012.

[195] S Aparna. Sanskrit to english translator. *Language in India*, 5, 2005.

[196] Promila Bahadur, A Jain, and DS Chauhan. English to sanskrit machine translation. In *Proceedings of the International Conference & Workshop on Emerging Trends in Technology*, pages 641–645. ACM, 2011.

[197] Vesselin Kiyurkchiev, Nikolay Pavlov, and Asen Rahnev. Cloud-based architecture of dispel. *International Journal of Pure and Applied Mathematics*, 120(4):573–581, 2019.

[198] Yanping Huang, Youlong Cheng, Ankur Bapna, Orhan Firat, Dehao Chen, Mia Chen, HyoukJoong Lee, Jiquan Ngiam, Quoc V Le, Yonghui Wu, et al. Gpipe: Efficient training of giant neural networks using pipeline parallelism. In *Advances in Neural Information Processing Systems*, pages 103–112, 2019.

[199] Yao Chen, Jiong He, Xiaofan Zhang, Cong Hao, and Deming Chen. Cloud-dnn: An open framework for mapping dnn models to cloud fpgas. In *Proceedings of the 2019 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*, pages 73–82, 2019.

[200] Ashish Vaswani, Samy Bengio, Eugene Brevdo, Francois Chollet, Aidan N Gomez, Stephan Gouws, Llion Jones, Łukasz Kaiser, Nal Kalchbrenner, Niki Parmar, et al. Tensor2tensor for neural machine translation. *arXiv preprint arXiv:1803.07416*, 2018.

[201] Ashish Venugopal and Andreas Zollmann. Grammar based statistical mt on hadoop: An end-to-end toolkit for large scale pscfg based mt. *The Prague Bulletin of Mathematical Linguistics*, 91:67–78, 2009.

[202] Jorge Ferrández-Tordera, Sergio Ortiz-Rojas, and Antonio Toral. Cloudlm: a cloud-based language model for machine translation. *The Prague Bulletin of Mathematical Linguistics*, 105(1):51–61, 2016.

[203] Rashid Ahmad, AK Rathaur, B Rambabu, Pawan Kumar, Mukul K Sinha, and Rajeev Sangal. Provision of a cache by a system integration and deployment platform to enhance the performance of compute-intensive nlp applications. In *African Conference on Software Engineering Applied Computing*, 2011.

[204] Pawan Kumar, Rashid Ahmad, Banshi Chaudhary, and Mukul Sinha. Dashboard: A tool for integration, validation, and visualization of distributed nlp systems on heterogeneous platforms. *The Companion Volume of the Proceedings of IJCNLP 2013: System Demonstrations*, pages 9–12, 2013.

[205] Rashid Ahmad, Pawan Kumar, B Rambabu, Phani Sajja, Mukul K Sinha, and Rajeev Sangal. Enhancing throughput of a machine translation system using mapreduce framework: An engineering approach. *ICON*, 2011.

[206] Pawan Kumar, Rashid Ahmad, BD Chaudhary, and M Sinha. An approach to assure qos of machine translation system on cloud. In *Procedings of The Fourth International Conference on Cloud Computing, GRIDs, and Virtualization*, pages 179–184, 2013.

[207] Andrejs Vasiļjevs, Raivis Skadiņš, and Jörg Tiedemann. Letsmt!: Cloud-based platform for building user tailored machine translation engines. *Proceedings of the 13th Machine Translation Summit*, pages 507–511, 2011.

[208] Rashid Ahmad, Pawan Kumar, Ashutosh Kumar, Mukul K Sinha, and BD Chaudhary. Improve user experience on web for machine translation system using storm. In *IEEE Fourth International Conference on Big Data and Cloud Computing*, pages 243–248. IEEE, 2014.

[209] Raivis Skadiņš, Jörg Tiedemann, et al. Letsmt!: A cloud-based platform for do-it-yourself machine translation. In *Proceedings of the ACL 2012 System Demonstrations*, pages 43–48. Association for Computational Linguistics, 2012.

[210] Qin Gao and Stephan Vogel. Training phrase-based machine translation models on the cloud: Open source machine translation toolkit chaski. *The Prague Bulletin of Mathematical Linguistics*, 93:37–46, 2010.

[211] Sriram Chaudhury, Ankitha Rao, and Dipti M Sharma. Anusaaraka: An expert system based machine translation system. In *Natural Language Processing and Knowledge Engineering (NLP-KE), 2010 International Conference on*, pages 1–6. IEEE, 2010.

[212] Amba Kulkarni. Samsaadhani, a sanskrit computational toolkit, 2002.

[213] Aasish Pappu and Ratna Sanyal. Vaakkriti: Sanskrit tokenizer. In *Proceedings of the Third International Joint Conference on Natural Language Processing: Volume-II*, 2008.

[214] Kumar Sachin. Sandhi splitter and analyzer for sanskrit (with reference to ac sandhi). *Mphil degree at SCSS, JNU (submitted, 2007)*, 2007.

[215] Anil Kumar, Vipul Mittal, and Amba Kulkarni. Sanskrit compound processor. In *Sanskrit Computational Linguistics*, pages 57–69. Springer, 2010.

[216] Sachin Kumar. Sandhi splitter and analyzer for sanskrit. *with special reference to aC sandhi*, 2007.

[217] Akshar Bharati, Amba P Kulkarni, and V Sheeba. Building a wide coverage sanskrit morphological analyser: A practical approach. In *The First National Symposium on Modelling and Shallow Parsing of Indian Languages, IIT-Bombay*, 2006.

[218] Vipul Mittal. Automatic sanskrit segmentizer using finite state transducers. In *Proceedings of the ACL 2010 Student Research Workshop*, pages 85–90. Association for Computational Linguistics, 2010.

[219] Girish Nath Jha, Muktanand Agrawal, Sudhir K Mishra, Diwakar Mani, Diwakar Mishra, Manji Bhadra, Surjit K Singh, et al. Inflectional morphology analyzer for sanskrit. In *Sanskrit computational linguistics*, pages 219–238. Springer, 2009.

[220] Amba Kulkarni, Sheetal Pokar, and Devanand Shukl. Designing a constraint based parser for sanskrit. In *Sanskrit Computational Linguistics*, pages 70–90. Springer, 2010.

[221] Pawan Goyal, Vipul Arora, and Laxmidhar Behera. Analysis of sanskrit text: Parsing and semantic relations. In *Sanskrit Computational Linguistics*, pages 200–218. Springer, 2009.

[222] Amba Kulkarni and Anil Kumar. Statistical constituency parser for sanskrit compounds. *Proceedings of ICON*, 2011.

[223] Amba Kulkarni and KV Ramakrishnamacharyulu. Parsing sanskrit texts: Some relation specific issues. In *Proceedings of the 5th International Sanskrit Computational Linguistics Symposium. DK Printworld (P) Ltd*, 2013.

[224] Amba Kulkarni. A deterministic dependency parser with dynamic programming for sanskrit. In *Proceedings of the Second International Conference on Dependency Linguistics (DepLing 2013)*, pages 157–166, 2013.

[225] Gérard Huet. Shallow syntax analysis in sanskrit guided by semantic nets constraints. In *Proceedings of the 2006 international workshop on Research issues in digital libraries*, page 6. ACM, 2006.

[226] Oliver Hellwig. Sanskrittagger: A stochastic lexical and pos tagger for sanskrit. In *Sanskrit Computational Linguistics*, pages 266–277. Springer, 2009.

[227] Oliver Hellwig. Performance of a lexical and pos tagger for sanskrit. In *Sanskrit Computational Linguistics*, pages 162–172. Springer, 2010.

[228] Murali Nandi and RJ Ramasree. Rule-based extraction of multi-word expressions for elementary sanskrit texts. *International Journal*, 3(11), 2013.

[229] Anil Kumar, V Sheebasudheer, and Amba Kulkarni. Sanskrit compound paraphrase generator. *Proceedings of ICON*, 2009.

[230] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: A system for large-scale machine learning. In *OSDI*, volume 16, pages 265–283, 2016.

[231] Nikhil Ketkar. Introduction to keras. In *Deep learning with Python*, pages 97–111. Springer, 2017.

[232] Kangkang Sun, Shaoshuai Mou, Jianbin Qiu, Tong Wang, and Huijun Gao. Adaptive fuzzy control for nontriangular structural stochastic switched nonlinear systems with full state constraints. *IEEE Transactions on Fuzzy systems*, 27(8):1587–1601, 2018.

[233] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

[234] Rico Sennrich and Barry Haddow. Linguistic input features improve neural machine translation. *arXiv preprint arXiv:1606.02892*, 2016.

[235] Yuanzhi Li and Yang Yuan. Convergence analysis of two-layer neural networks with relu activation. In *Advances in Neural Information Processing Systems*, pages 597–607, 2017.

[236] Guido F Montufar, Razvan Pascanu, Kyunghyun Cho, and Yoshua Bengio. On the number of linear regions of deep neural networks. In *Advances in neural information processing systems*, pages 2924–2932, 2014.

[237] Sanskrit News. Sanskrit News,"department of public health relations", 2018. Accessed on: 13-1-2019.

[238] Yonatan Belinkov and Yonatan Bisk. Synthetic and natural noise both break neural machine translation. *arXiv preprint arXiv:1711.02173*, 2017.

[239] JNU Computational Linguistics. Sanskrit corpora, 2019. Accessed on: 19-10-2019.

[240] University of Hyderabad Department of Sanskrit Studies. Digital sanskrit corpora, 2019. Accessed on: 19-10-2019.

[241] Matthew D Zeiler. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*, 2012.

[242] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[243] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[244] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics, 2002.

[245] Tomáš Mikolov, Martin Karafiát, Lukáš Burget, Jan Černockỳ, and Sanjeev Khudanpur. Recurrent neural network based language model. In *Eleventh annual conference of the international speech communication association*, 2010.

[246] Samsaadhaniisanskrit to hindi accessor cum machine translator, 2016. Accessed: 2016-09-08.

[247] Rajneesh Pandey, Atul Kr Ojha, and Girish Nath Jha. Demo of sanskrit-hindi smt system. *arXiv preprint arXiv:1804.06716*, 2018.

[248] Gérard Huet, Amba Kulkarni, and Peter Scharf. Sanskrit computational linguistics. *Lecture Notes in Computer Science*, 5402, 2009.

[249] Ling Qian, Zhiguo Luo, Yujian Du, and Leitao Guo. Cloud computing: An overview. In *IEEE International Conference on Cloud Computing*, pages 626–631. Springer, 2009.

[250] EC Amazon. Amazon web services, 2020. Accessed on: 28-03-2020.

[251] Yuping Xing and Yongzhao Zhan. Virtualization and cloud computing. In *Future Wireless Networks and Information Systems*, pages 305–312. Springer, 2012.

[252] Adimugan Kumaran and Tobias Kellner. A generic framework for machine transliteration. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 721–722. ACM, 2007.

[253] Kevin Knight and Jonathan Graehl. Machine transliteration. *Computational linguistics*, 24(4):599–612, 1998.

[254] Nicola Bertoldi, Mauro Cettolo, and Marcello Federico. Statistical machine translation of texts with misspelled words. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 412–419, 2010.

[255] Wei Wang, Kevin Knight, and Daniel Marcu. Capitalizing machine translation. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, pages 1–8, 2006.

[256] Mary Flanagan. Error classification for mt evaluation. In *Technology Partnerships for Crossing the Language Barrier: Proceedings of the First Conference of the Association for Machine Translation in the Americas*, pages 65–72, 1994.

[257] Lucian Vlad Lita, Abe Ittycheriah, Salim Roukos, and Nanda Kambhatla. Truecasing. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*, pages 152–159. Association for Computational Linguistics, 2003.

[258] Akakpo Agbago, Roland Kuhn, and George Foster. Truecasing for the portage system. In *Recent Advances in Natural Language Processing*, pages 21–24. Citeseer, 2005.

[259] John Lafferty, Andrew McCallum, and Fernando CN Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. 2001.

[260] Andrew M Dai, Klaus Macherey, Franz Josef Och, Ashok C Popat, and David R Talbot. Compound splitting, July 7 2015. US Patent 9,075,792.

[261] Amba Kulkarni and Anil Kumar. Clues from as. t. adhyayı for compound type identification. In *Proceedings of the International Sanskrit Computational Linguistics Symposium. DK Printworld (P) Ltd*, 2013.

[262] Amba Kulkarni, Soma Paul, Malhar Kulkarni, Anil Kumar Nelakanti, and Nitesh Surtani. Semantic processing of compounds in indian languages. In *Proceedings of COLING 2012*, pages 1489–1502, 2012.

[263] Panagiotis Karageorgakis, Alexandros Potamianos, and Ioannis Klasinas. Towards incorporating language morphology into statistical machine translation systems. In *IEEE Workshop on Automatic Speech Recognition and Understanding, 2005.*, pages 80–85. IEEE, 2005.

[264] Oliver Hellwig, Salvatore Scarlata, Elia Ackermann, and Paul Widmer. The treebank of vedic sanskrit. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 5137–5146, 2020.

[265] Francisco Guzmán, Ahmed Abdelali, Irina Temnikova, Hassan Sajjad, and Stephan Vogel. How do humans evaluate machine translation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 457–466, 2015.

[266] Matthew Snover, Nitin Madnani, Bonnie J Dorr, and Richard Schwartz. Fluency, adequacy, or hter?: exploring different human judgments with a tunable mt metric. In *Proceedings of the Fourth Workshop on Statistical Machine Translation*, pages 259–268. Association for Computational Linguistics, 2009.

[267] Eduard Hovy, Chin-Yew Lin, et al. Automated text summarization in summarist. *Advances in automatic text summarization*, 14, 1999.

[268] Martin Rajman and Tony Hartley. Automatically predicting mt systems rankings compatible with fluency, adequacy or informativeness scores. In *Proceedings of the Workshop on Machine Translation Evaluation:"Who Did What To Whom*, pages 29–34, 2001.

[269] Rita Nübel. End-to-end evaluation in verbmobil i. *Proceedings of MT Summit VI*, pages 232–239, 1997.

[270] Sonja Nießen, Franz Josef Och, Gregor Leusch, Hermann Ney, et al. An evaluation tool for machine translation: Fast evaluation for mt research. In *LREC*, 2000.

[271] Joseph P Turian, Luke Shea, and I Dan Melamed. Evaluation of machine translation and its evaluation. Technical report, NEW YORK UNIV NY, 2006.

[272] Eduard Hovy, Maghi King, and Andrei Popescu-Belis. An introduction to mt evaluation. In *Proceedings of Machine Translation Evaluation: Human Evaluators meet Automated Metrics. Workshop at the LREC 2002 Conference. Las Palmas, Spain*, pages 1–7, 2002.

[273] Joost CF de Winter, Samuel D Gosling, and Jeff Potter. Comparing the pearson and spearman correlation coefficients across distributions and sample sizes: A tutorial using simulations and empirical data. *Psychological methods*, 21(3):273, 2016.

[274] Satanjeev Banerjee and Alon Lavie. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72, 2005.

Appendix

A.1 Performance Evaluation

Performance of the system is evaluated by manual and automatic mode. The manual evaluation method performs the subjective evaluation on the output of MTS while the automatic performs the objective evaluation.

A.1.1 Manual Performance Evaluation

It is referred to as subjective criteria of evaluation as two evaluators evaluating the same text may have different rating for the translated text. This method is the best, but it requires a tremendous amount of time and cost though not reusable. Humans use their world, cultural and context knowledge for analysis of source language for extracting information contained in the language. Humans evaluate the translated text based on several measures. The most frequently used method for evaluation is conducted by linguist annotating a tag corresponding to each error occurred. The evaluators are provided with the source text, machine-translated output and reference translation. The scores are aligned by the annotator between 1 to 5 based on reference translation corresponding to the source sentence. The overall quality of the human evaluation depends on the different parameters as experimented by [265] on Spanish-English Translation. The results display that bilingual evaluators perform task faster than monolingual as been provided with the correct reference translation but monolingual evaluators are more compatible. The method of eye-tracking is more time-consuming and requires rich linguistic knowledge.

1. Fluency It is referred to the degree at which translated text follows target language grammar. The output translation English fluency is examined involving several measures such as grammatical correctness, understandability, readability, and idiomatic word choices. Various methods have been proposed to measure the fluency such as rating the sentence by human evaluators (judges) based on a scale, based on syntactic structure[266]. Fluency Scale varies from 5-0 as follows: Flawless-(5), Good-(4) Non-Native-(3) Not Fluent-(2) Incomprehensible-(1).

2. Adequacy This measure is used to examine the amount of information of source text is contained in the translated target text[267][268]. Adequacy Scale: All meaning-(5) Most meaning-(4) Much meaning-(3) Little meaning-(2) None-(1)

3. Fidelity It refers to the amount of information carried in the translated text in comparison to reference translation[269][270].

4. Precision and Recall Assume that Y is machine translated output of MTS and X is reference translation, whereas in the numerator we have the intersection of machine translation and reference translation as in Eq.(1-2). The problem of this measure

is to compute intersection. This measure is also used for other NLP task where reference translation is given (such as summarization, Text generation)[271].

$$P\frac{Y}{X} = \frac{|X \cap Y|}{|Y|} \tag{6.1}$$

$$R\frac{Y}{X} = \frac{|X \cap Y|}{|X|} \tag{6.2}$$

5. Understandability It states that even if some translated text is completely correct, it may not be understandable to some target user[272].

A.1.2 Automatic error evaluation

Automatic MT evaluation was formed to overcome the drawback of the manual method. As the manual evaluation criteria incur high cost and time using subjective metrics. These evaluation criteria deprive in repeatability and automatic tuning of MTS[74]. It evaluates the quality of MTS by comparing the MT output with the reference translations. It's measured in terms of correlational scores[273] determined by fluency and adequacy of the output. The different automatic evaluation metrics are listed in Table 6.1.

Table 6.1: Evaluation Measures

Evaluation Metric	Formula	Baseline Values	Notions
BLEU [244]	$BLEU = min(1, \frac{output\_length}{Reference\_length})(\prod_{i=1}^{4} precision_i)$ $P_n = \frac{\sum_{c \in candidates} \sum_{n-gram \in c} Count_{clip}(n-gram)}{\sum_{c' \in (candidate)} \sum_{n-gram \in c'} Count(n-gram')}$ $BP = \left\{ \begin{array}{c} 1 \text{ if } c > r \\ e^{1-\frac{r}{c}} \text{ if } \quad c < r \end{array} \right\}$ $BLEU = BP \times \exp\left(\Sigma_{n=1}^{N} W_n \log p_n\right)$ $\log \text{BLEU} = \min\left(1 - \frac{r}{c}, 0\right) + \sum_{n=1}^{N} w_n \log p_n$	$N = 4$ $w_n = 1/N$	c= total length of candidate translation corpus r=test corpus sufficient reference length n=number of words in n-gram (consecutive words form n-gram up to length).
METEOR [274]	Fmean $= \frac{10PR}{R+9P}$ $Penalty = 0.5 \times (\frac{numberofchunks}{numberofunigram_matched})^3$ Score = Fmean $\times$ (1 $-$ Penalty)	Score range=0 to 5	P= Precision R=Recall for MTS output and reference translation.
WER[270]	$WER = \frac{s+l+D}{N}$	Range= 0 (if all words are same) Range=1 (if all words are different)	S=Substitutions, I=Insertions, D=Deletions
F-measure	$Precision(Candidate\|Reference) = \frac{\|Reference \cap Candidate\|}{Candidate}$ $Recall(Candidate\|Reference) = \frac{\|Reference \cap Candidate\|}{Reference}$ $F - Measure = \frac{2 \times P \times R}{P+R}$	Harmonic Mean of Precision and Recall	P=Precision R=Recall

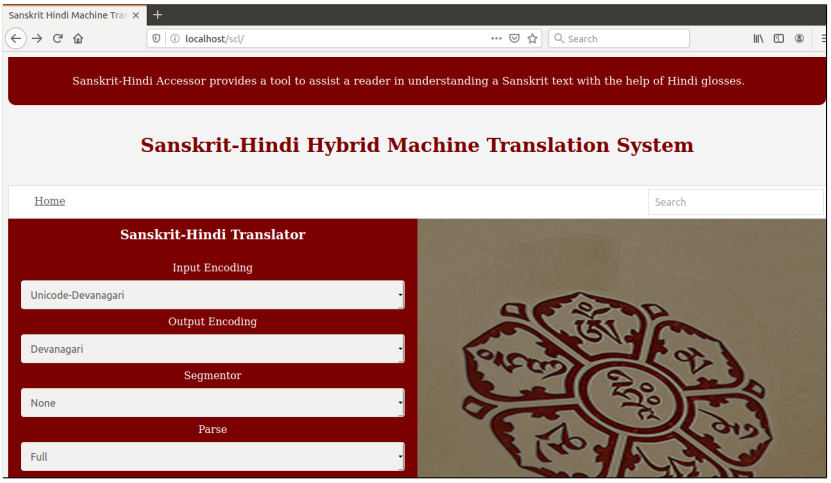
Table 6.2: Library Used in the Research Work

Library	**Description**
Numpy	It is a python library used as multi-dimensional arrays and matrices. It is used to write or arrange data in tabular format.
Pandas	It is an open-source python library used to handle various datasets. Easy to handle missing values or we can say Nan values in data. It can easily perform operations like insertion, deletion using pandas. Contains features like Automatic and explicit data alignment. It is easy to assign labels or use labelled values using pandas.
Sklearn	It is the python library used for data mining and data analysis. It is designed to make an interoperate with python libraries like NumPy, script, matplotlib etc.
Tensor Flow	Tensor flow is an open-source software python library used to design and build various deep learning models. It is also used for numerical computations using data flow graphs. These graphs contain nodes which represent mathematical operations and edges that represent data that is communicated between edges.
KERAS	Keras is a minimalist neural network library in python. It contains various neural building blocks like layers, optimizers, activation functions etc. It is designed to make deep learning models faster and easier as much as possible.
RNN	It is a part of an artificial neural network which is used for language models mostly language model mainly consists of two folds. First one is arbitrary sentences how they occur in reality which helps us to measure grammatical and semantic correctness the Second one is to generate a new text.

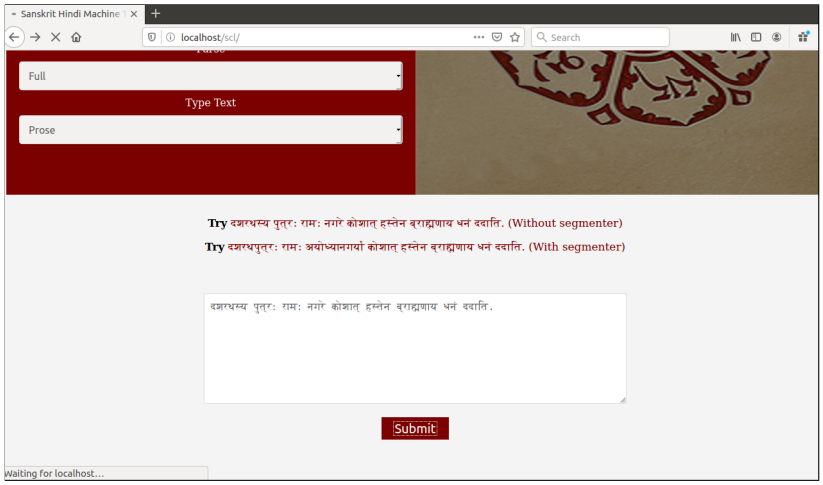
A.2 Web interface for *Sanskrit- Hindi Hybrid Machine Translation System*

A web-based application is developed for translating Sanskrit-Hindi using linguistic tools and neural-based. Here is the demonstration of a web application for the translation of Sanskrit to Hindi language using the hybrid approach.

1. In the first step, the user can view the HTML page of Sanskrit-Hindi Machine Translation System.

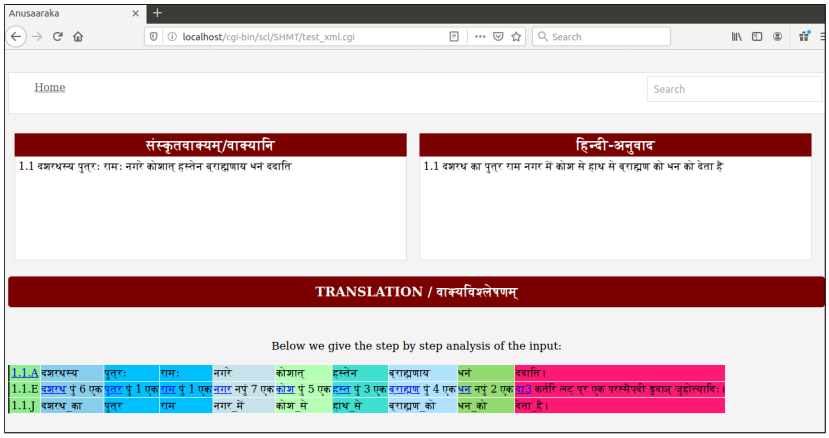


2. Type or paste Sanskrit sentence to be translated and click on submit.

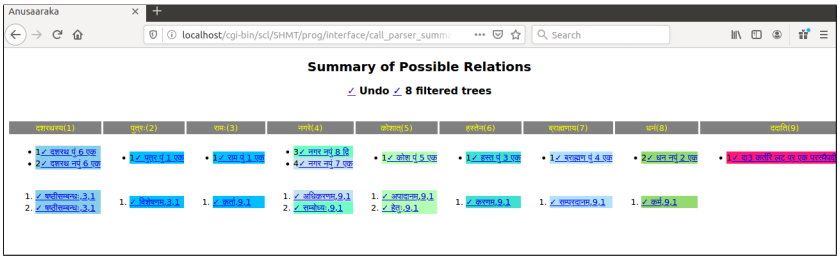


3. User can view the Sanskrit and Hindi translation in parallel window.

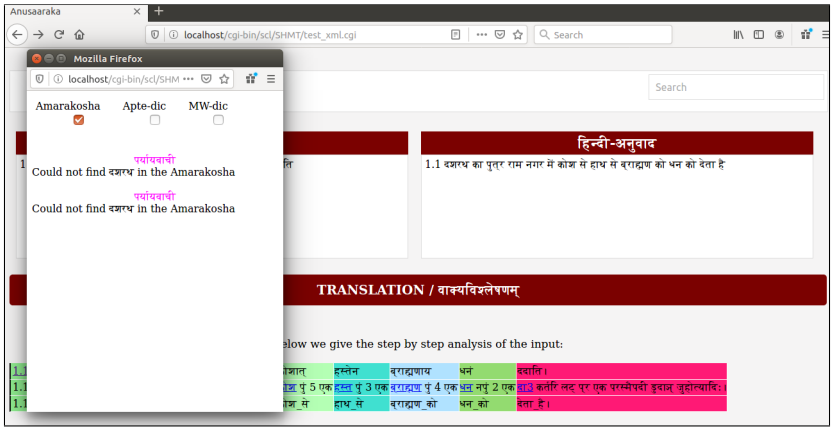
4. A ellobrate view of translation using Anusaaraka engine is shown at the bottom of the page.



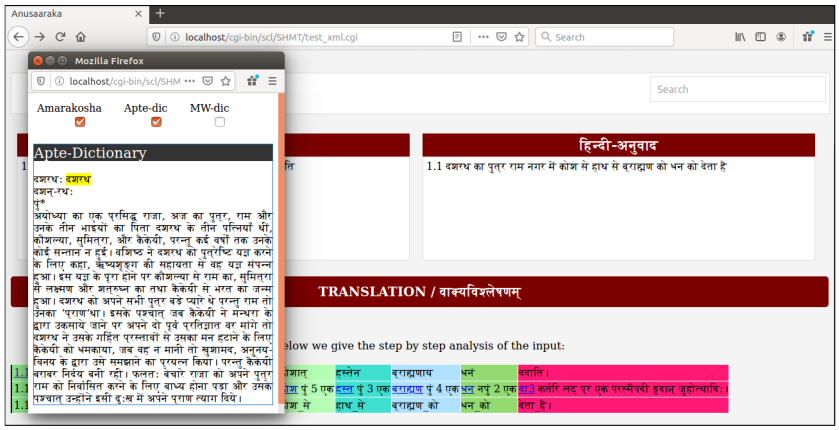
5. User will get the summary of the possible relations of the submitted sentences in detailed view.



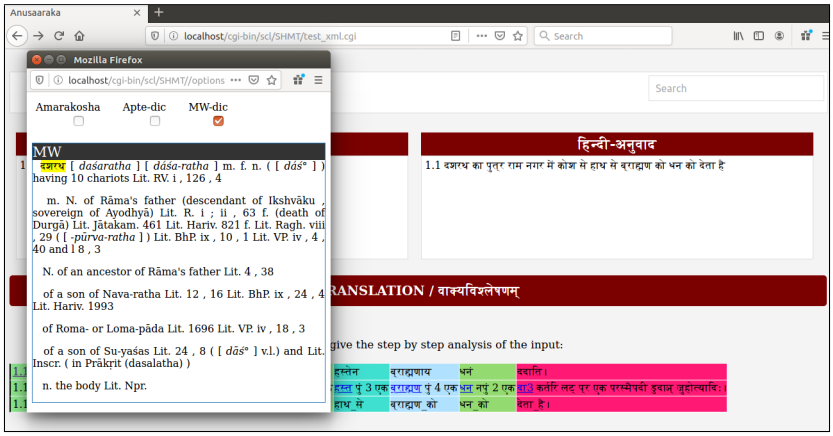
6. User can also view the dictionary meaning of words of the submitted sentence from Amarakosha.



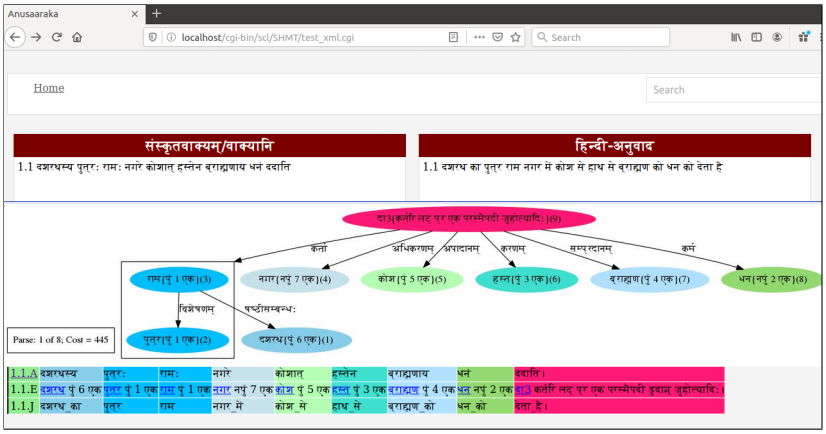
7. User can also view the dictionary meaning of words of the submitted sentence from Apte-dic.



8. User can also view the dictionary meaning of words of the submitted sentence from Monier-William(MW-Dic).



9. Finally entire sentence can be viewed using parse tree corresponding to grammatical categories.


Anusaaraka
localhost/cgi-bin/scl/SHMT/test_xml.cgi
Search
Home
Search
संस्कृतवाक्यम्/वाक्यानि
1.1 दशरथस्य पुत्रः रामः नगरे कोशात् हस्तेन ब्राह्मणाय धनं ददाति
हिन्दी-अनुवाद
1.1 दशरथ का पुत्र राम नगर में कोश से हाथ से ब्राह्मण को धन को देता है
दा3(कर्तरि लट् पर एक परस्मैपदी जुहोत्यादिः)(0)
कर्ता
अधिकरणम्
अपादानम्
करणम्
सम्प्रदानम्
कर्म
राम(पुं 1 एक)(3)
नगर(नपुं 7 एक)(4)
कोश(पुं 5 एक)(5)
हस्त(पुं 3 एक)(6)
ब्राह्मण(पुं 4 एक)(7)
धन(नपुं 2 एक)(8)
विशेषणम्
षष्ठीसम्बन्धः
Parse: 1 of 8; Cost = 445
पुत्र(पुं 1 एक)(2)
दशरथ(पुं 6 एक)(1)
1.1.A दशरथस्य पुत्रः रामः नगरे कोशात् हस्तेन ब्राह्मणाय धनं ददाति।
1.1.E दशरथ पुं 6 एक पुत्र पुं 1 एक राम पुं 1 एक नगर नपुं 7 एक कोश पुं 5 एक हस्त पुं 3 एक ब्राह्मण पुं 4 एक धन नपुं 2 एक दा3 कर्तरि लट् पर एक परस्मैपदी जुहोत्यादिः।
1.1.J दशरथ_का पुत्र राम नगर_में कोश_से हाथ_से ब्राह्मण_को धन_को देता_है।

Chapter 7

List of Publications

SCI-Indexed Journals

- SCI-Indexed Journal[Published]

  1. Singh M, Kumar R, Chana I. Machine Translation Systems for Indian Languages: Review of Modeling Techniques, Challenges, Open Issues and Future Research Directions. Archives of Computational Methods in Engineering. 2020 Jun 17:1-29.[**Impact factor: 7.3**]
  2. Singh, M., Kumar, R. & Chana, I. Improving neural machine translation for low-resource Indian languages using rule-based feature extraction.Neural Computing & Application(2020). https://doi.org/10.1007/s00521-020-04990-9 [**Impact Factor: 4.7**]
  3. Singh, M., Kumar, R. & Chana, I. A forefront to machine translation technology: deployment on the cloud as a service to enhance QoS parameters.Soft Computing(2020). https://doi.org/10.1007/s00500-020-04923-7[**Impact factor: 3.0**]

- SCI-Indexed Journal[Under-Review]

  1. Muskaan Singh, Ravinder Kumar, and Inderveer Chana. "Machine Translation Intelligent System as a Service: An aid to Teaching and Learning Process" ,Computer, IEEE Computer Society.[SCI,IF=3.5][Major Revision]
  2. Muskaan Singh, Ravinder Kumar, and Inderveer Chana. "A Error Taxonomy for Morphologically-Rich Languages: A Case Study on Sanskrit-Hindi language", Transaction for Asian and Low Resource Language Information Processing, ACM [SCI, IF 2.2]

Scopus-Indexed Journals

1. Muskaan Singh, Ravinder Kumar, and Inderveer Chana, "Hybrid Machine Translation System Using Deep Learning". ASM Sc. J., 13, Special Issue 2, 2020 for ICSCC2019, 31-45

2. Muskaan Singh, Ravinder Kumar, and Inderveer Chana, "Corpus based Machine Translation System with Deep Neural Network for Sanskrit to Hindi Translation". Procedia Computer Science 167 (2020): 2534-2544.

International Conference

1. Muskaan Singh, Ravinder Kumar, and Inderveer Chana, "Improving Neural Machine Translation Using Rule-Based Machine Translation". 7th International Conference on Smart Computing & Communications (ICSCC),Curtin University, Miri, Malaysia, IEEE, 2019[**BEST PAPER AWARD**]

2. Muskaan Singh, Ravinder Kumar, and Inderveer Chana, "Neural Machine Translation for Low-Resource Languages". ACM, GHC-Poster Presentation, Orlando,USA,2019.

3. Muskaan Singh , Ravinder Kumar, and Inderveer Chana, "Neuro-FGA Based Machine Translation System for Sanskrit to Hindi Language". International Conference on Innovative Sustainable Computational Technologies (CISCT), Dehradun, Uttarakhand, India. IEEE, 2019.

4. Muskaan Singh, Ravinder Kumar, and Inderveer Chana, "Neural-Based Machine Translation System Outperforming Statistical Phrase-Based Machine Translation for Low-Resource Languages". Twelfth International Conference on Contemporary Computing (IC3), Noida, India, IEEE, 2019.

5. Muskaan Singh, Ravinder Kumar, and Inderveer Chana, "Encoding-Decoding Methods for Neural Machine Translation". 2nd International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT),Kerala, India, Vol. 1. IEEE, 2019.

6. Muskaan Singh, Ravinder Kumar, and Inderveer Chana, "GA-based machine translation system for Sanskrit to Hindi language". Recent Trends in Communication, Computing, and Electronics. Springer, Singapore, 2019. 419-427.

7. Muskaan Singh, Ravinder Kumar, and Inderveer Chana, "Neural Machine Translation for Low-Resource Languages" ACM, GHCI-Poster Presentation, Banglore, India ,2019.[**2nd Prize in Poster Presentation**]

8. Muskaan Singh, Ravinder Kumar, and Inderveer Chana. "CDFM-based Secure & Efficient Architecture for Data Management in Cloud Computing." 2019 2nd International Conference on Intelligent Communication and Computational Techniques (ICCT). IEEE, 2019.