Neural Computing and Applications
https://doi.org/10.1007/s00521-020-05156-3

ORIGINAL ARTICLE

A Sanskrit-to-English machine translation using hybridization of direct and rule-based approach

Sitender[1,2] · Seema Bawa[1]

Received: 21 January 2019 / Accepted: 19 June 2020

Abstract
The work in this paper presents a MTS from Sanskrit to English language using a hybridized form of direct and rule-based machine translation technique. This paper also discusses the language divergence among Sanskrit and English languages with a recommended solution to handle the divergence. The proposed system has used two bilingual dictionaries (Sanskrit–English, Sanskrit–UNL), a tagged Sanskrit corpus, a Sanskrit analysis rule base and an ELGR base. Elasticsearch technique has enhanced the translation speed of the proposed system for accessing the data from different data dictionaries and rule bases used for the system development. The system uses CFG in CNF for Sanskrit language processing and CYK parsing technique for processing the input Sanskrit sentence. This work also presents a novel algorithm which creates a parse tree from the parsing table. ELGR base and bilingual dictionaries generate the target language sentence. The proposed system is evaluated using natural language toolkit API in python and achieved a BLEU score of 0.7606, fluency score of 3.63 and adequacy score of 3.72. A comparison of the proposed system with state-of-the-art systems shows that the proposed system outperforms existing systems.

Keywords Machine translation · Sanskrit language grammar · CYK parser · BLEU score · Elasticsearch · POS tagging · Nltk

Abbreviations

AI	Artificial intelligence
API	Application programming interface
BLEU	Bilingual evaluation understudy
CBMT	Corpus-based machine translation
CFG	Context-free grammar
CLR	Canonical syntactic realization
CNF	Chomsky normal form
CYK	Cocke–Younger–Kasami
DMT	Direct machine translation
GLR	Generalized linking routine
HBMT	Hybrid-based machine translation
LCS	Lexical conceptual structure
MT	Machine translation
MTS	Machine translation system
POS	Part of speech
RBMT	Rule-based machine translation
TLGR	Target language generation rule
UNL	Universal networking language

✉ Sitender
Sitender@thapar.edu; Sitender@msit.in

Seema Bawa
Seema@Thapar.edu

[1] Computer Science and Engineering, Thapar Institute of Engineering and Technology, Patiala, India

[2] MSIT, New Delhi 110058, India

1 Introduction

Sanskrit means "Samaskrita," i.e., a decorated, refined and purified language written in Devanagari script. Sanskrit is one of the oldest languages in the world, and all other Indian languages have originated from Sanskrit. The size of Sanskrit corpus is over 30 million, which is hundred times bigger than the combined corpus of both Greek and Latin languages and was available even before printing press came into existence in the form of inscriptions. Sanskrit text consists of poetic, dramatic, devotional, scientific, engineering, mathematical and literary text. The Sanskrit language is a highly inflected language in nature [1] represented the Sanskrit language in a better structured and easy-to-understand way in the form of eight

Published online: 17 July 2020

chapters (Panini Asthadhyayi). Techniques for translating text using computers date back to Georges Artstrouni' mechanical brain (1937), and high reliability, robustness and the efficiency of current computer-based translation techniques have lead to the development of commercial level products. The census of India (1991) has shown that the speakers of Sanskrit language in India were only 49,736. This number is expected to reduce in the future, and Sanskrit scholar may become hard to find, which would create a more difficult situation to understand the ancient Indian text available in Sanskrit. Recently, numerous steps have been taken by various government and non-government organizations at national as well as international level to save this ancient language (Sanskrit) and uncover the science and technology written in this language for the betterment of humanity, especially in medical sciences. According to [2], Sanskrit language due to its most structured and mathematically organized form of grammar as compared to other natural languages could create a new milestone in the field of AI research and could be used as a better language for computer understanding than other natural languages.

English has emerged as a most used language in the world with the invention of the World Wide Web. So converting an ancient language, i.e., Sanskrit, into a modern language like English has become much desired yet remains to be a challenging task. MT is a mechanism for translating text from one language to another language through computers. MT technique can resolve this problem of translating Sanskrit text into English-equivalent text efficiently and easily. Although some significant work has been done in English to Sanskrit MTS development by [3–6], there is lot of scope in Sanskrit-to-English MT system development. Human translation is not the core solution to this problem.

1.1 Uses and benefits of the MTS

According to Ethnologue Languages of World, approximately 7102 languages and thousands of dialects have been used by people in the world [7]. Human translation has never been an effective solution for such problems due to less availability of human translators, high cost of manual translation and difficulty to approach by everyone. According to Census of India 2001 data, 22 scheduled and 100 nonscheduled languages with approximately 1600 local dialects are being used by people [8, 9]. So for the development of country like India, people have to exchange technology, science, ideas and work together without any language barrier. MT techniques can remove such problems in an effective manner. So there is a great need for MT at the global level as well as local level in India also. MTS in general has its uses in every field of life; some of them are tourism, health domain, finance, defense, education, business, government work, Web content, app development. The proposed translation system could be used in teaching/learning of the Sanskrit language in schools, to understand the features of Sanskrit language (one of the most unambiguous languages, well-structured grammar, divine feature, best suited for computers as accepted by NASA, treasure of ancient science and technology, meditation power, rich in named entities) for research purpose. MTS has several benefits over the traditional methods of translation, which includes the high translation speed, less costly, more memory than human to remember large data, easy to translate into multiple languages at once in multilingual environment, translation could be done without any fatigue, availability of the system any time anywhere.

This paper is segmented into seven sections. Section 1 gives the introduction about Sanskrit language and need for machine translation for Sanskrit to English language. Section 2 discusses literature review of various machine translation approaches and MTS developed by different researchers. Section 3 describes various language divergences which occur during translation from Sanskrit and English language with recommendations to handle them. Section 4 describes the proposed Sanskrit-to-English machine translation system having six modules. Section 5 gives the details of data dictionaries, rule bases, tagged corpus and technology used for implementing the proposed system. Section 6 describes the evaluation methods used for evaluating the proposed system and comparing with other existing systems. Section 7 gives an informative conclusion of the article followed by references.

2 Review of the existing MT system

Before starting the process of translation between any language pair, the study of language divergence is an important step. According to [10–12], language divergence may occur at syntactic level, thematic level, inflation level, lexical level, particle level, gerund level and voice level. For processing the natural language text through the computer, Chomsky hierarchy of grammar has been used commonly. To parse the input sentence, a suitable parsing algorithm has been used. Several approaches have been used by researchers to develop MTS for different languages, and these approaches can be categorized into four groups as DMT, RBMT, CBMT and HBMT. Figure 1 shows the review of various MT systems developed based on different approaches for Indian languages. It is evident from Fig. 1 that the number of MTS developed using CBMT is higher than RBMT, which is in turn higher than HBMT and DMT.

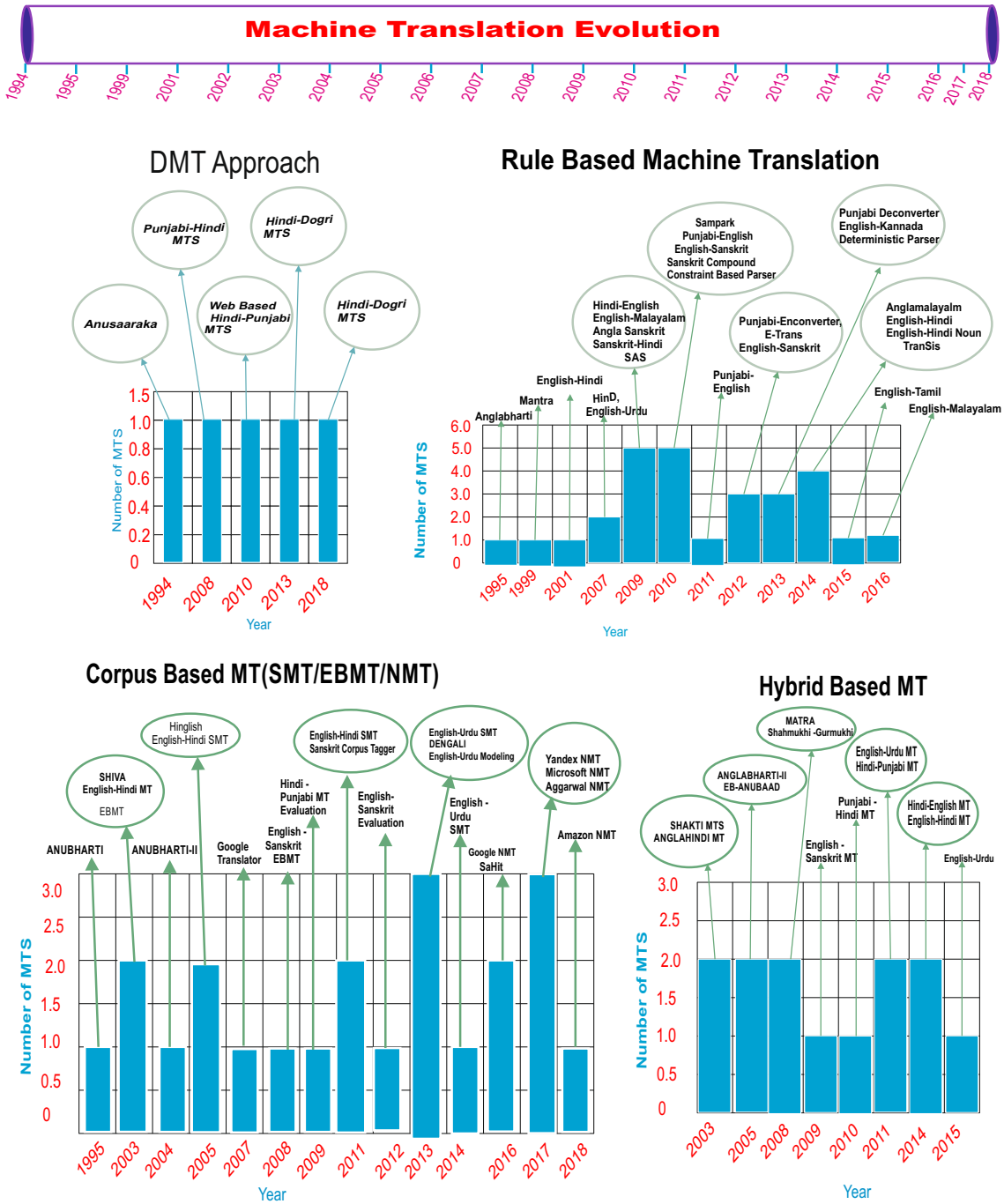

Fig. 1 Development of MT based on different approaches

DMT approach includes: [13–19] machine translation systems. This approach does not require syntactic or semantic analysis of the source or target language and so does not require huge effort in development. This approach is still in use for developing MTS rapidly. *RBMT* includes: [6, 20–39] MTS. This approach requires syntactic as well as semantic analysis of the source as well as target language for translation. Although it requires huge effort and time for development, the efficiency of MT system based on RBMT approach has been the highest than other approaches. *CBMT* includes: [4, 23, 40–56] MTS. This approach covers example-based, statistical-based and neural MT systems. With the increase in availability of language resources in digital form, computational algorithms and computing power, this approach has become the choice of MT developers.

HBMT includes: [5, 18, 57–59, 59–66] MTS.

Table 1 Language divergence among Sanskrit and English

No.	Divergence	Explanation	Example	Recommendations
1	Thematic divergence	It arises only in case of the logical subject. In Sanskrit to English, the thematic divergence occurs when the re-positioning of subject and object occurs concerning verb from Sanskrit to English translation.	(i) (a) SS: मह्यम् मधुर्म् रोचते(Mahyam Madhurm Rocate) (b) ES: Sweets are liked by me.	Re-positioning of argument with respect to given head.
2	Structural divergence	It occurs from Sanskrit to English when the realization of Noun in Sanskrit is done with vibhakti (preposition) whereas in English the vibhakti part is realized with or without a preposition.	(i) (a) SS: अहम् फलम् खादामि (Aham Phalam Khaadaami) (b) ES:I eat fruits (ii) (a) SS: अहम् विद्यालयम् गच्छामि(Aham Vidyaalayam Gacchaami) (b) ES: I go to school (iii) (a) SS: रामः सीताम् ददर्श (Raamah Siitaam Dadarsha) (b) ES: Ram saw Sita. In sentence i(a),ii(a) and iii(a) the noun फलम् , विद्यालयम् and सीताम् are realized with vibhakti in Sanskrit but in English i(b),ii(b) and iii(b) their equivalents are realized with (to school) or without (fruits, Sita) preposition .	Such divergence can be resolved by using * marker in the lexicon for such items to indicate that corresponding elements in English may need compositional realization.
3	Inflation Divergence	From Sanskrit to English the Inflational divergence occurs in which one Sanskrit word gets translated into more than one English words.Example (i) and (ii) shows such divergence.	(i) (a) SS: रामः पन्डितायते(Raamah Panditaayate) (b) ES: Ram behaves like a scholar. (ii) (a) SS: सीता पापच्चयते(Siitaa Paapaccayate) (b) ES: Sita cooks again and again. (Sita) (cooks again and again)	Proper attention is required for handling the compound words of Sanskrit in this case.
4	Categorical Divergence	Categorical divergence occurs from Sanskrit to English Translation as shown in example (i).	(i) (a) SS: रामः सीताया ईर्ष्यति (Raamah Siitaayaa Iirssyati) (b) ES: Ram is jealous of Site. As seen in (i)(b) Jealous is recognized as verbal in POS tagging in Sanskrit whereas in English it is adjective.	It is related to CSR not with the GLR. This issue can be resolved using CAT parameter in Sanskrit lexical entry not in English as suggested by [10].
5	Lexical Divergence	Lexical divergence arises due to other divergences as shown in examples (i) and (ii).	(i) (a) SS: सः विद्यालयम् गच्छति (Sah Vidyaalayam Gacchati) (b) ES: He goes to school. (ii) (a) SS: सः विद्यालयात् आगच्छति (Sah Vidyaalayaat Aagacchati) (b) ES: He comes from school. As shown in example (i) and (ii) sentences for Sanskrit गच्छ is translated by English verb 'go', whereas in the next sentence Sanskrit verb 'आ +गच्छ' is translated by another verb 'come' which shows the lexical divergence among Sanskrit and English language.	Such divergence problem could be solved by proper selection of the lexical item.
6	Particle Divergence	In Sanskrit the participle is formed by adding the suffix '□□□'(तुम्) directly to any root verb and in English it is formed by adding 'to' before the verb. (i) In Sanskrit the dative case may be used in place of using 'तुमुन्' infinitive and replaces 'to' with 'for' in English sentence as shown in example (i),(ii),(iii) and (iv). (ii) Use of Particle 'sma'. When added to present tense it converts into Past tense as shown in example (v) and (vi). (iii) When ma sma मा स्म is added to past tense in Sanskrit it gets converted into present as shown in example (vi) and (vii).	(i) (a) SS: सीता वक्तुम् इच्छति (Siitaa Vaktum Icchati) (b) ES: Sita wants to speak. (ii) (a) SS: सीता वचनाय इच्छति (Siitaa Vacanaaya Icchati) (b) ES: Sita desires for speaking. (iii) (a) SS: नगरम् संरक्षितुम् रक्षकाः सन्ति (Nagaram Samrakssitum Raksakaah Santi) (b) ES: The police are to protect the city. (iv) (a) SS: नगरसंरक्षणाय रक्षकाः सन्ति (Nagarasamrakssannaaya Raksakaah Santi) (b) ES: The police is for the protection of the city. (v) (a) SS: अहम् खादामि (Aham Khaadaami) (b) ES: I am eating (vi) (a) SS:अहम् खादामि स्म (Aham Khaadaami Sma) (b) ES: I was eating (vii) (a) SS: त्वम् मूर्खम् अभवः (Tvam Muurkham Abhavah) (b) ES: You became a fool. (viii) (a) SS: त्वम् मूर्खम् मा स्म भवः (Tvam Muurkham Maa Sma Bhavah) (b) ES: Do not be a fool.	1. if SS contains 'tum'(तुम्) suffix then add 'to' before the corresponding verb in English for making infinitive particle or if SS contains dative case then add 'for' before the corresponding verb in English. 2. if (SS= verb + स्म sma) then use past tense in English else use the present tense. 3. if (SS=verb +मा स्म (maa sma)) then the verb gets converted into present tense.
7	Gerund Divergence	Gerund divergence occurs at the time of gerund realization in both Sanskrit and English Language. When the single subject is performing two tasks, then to show the completion of the first task before the commencement of the second one, we use 'क्त्वा or ल्यप्' past participles instead of using 'and then' phrase as shown in example (i) and (ii).	(i) (a) SS: बालकाः तेषां अभ्यासं कृत्वा विद्यालयम् गच्छन्ति(Baalakaah Tessaam Abhyaasam Krtvaa Vidyaalayam Gacchanti) (b) ES: Boys go to school having done their study. (ii) (a) SS: आगराम् गत्वा वयं ताजमहलम् द्रक्ष्यामः (Aagaraam Gatvaa Vayam Taajamahalam Drakssyaamah) (b) ES: Having gone to Agra, we will see the Tajmahal.	To resolve this divergence, whenever we see 'त्वा' □□ ल्यप्' ending with a verb,then use 'having + 3rd form of the verb' in English.

Table 1 continued

8	Voice divergence	In Sanskrit, there are three types of voices: Active (कर्तरि), Passive (कर्मणि) and Bhave (भावे) whereas in English we have only two voices: Active and Passive. Divergence is shown in example i,(ii) and (iii) sentences.	(i) (a) SS: भक्ता: देवीम् पूज्यन्ति (Bhaktaa: Deviim Puujyanti)(Active) (b) The devotees worship the goddess. (ii) (a) SS: भक्तै: देवी पूज्यते (Bhaktaih Devii Puujyate) (Passive) (b) The goddess is worshiped by the devotees. (iii) (a) SS: भक्तै: देव्या पूज्यते (Bhaktaih Devyaa Puujyate)(Bhave) (b) The goddess is being worshiped by the devotees.	To translate Bhave voice of Sanskrit in which subject and object both will be in an instrumental case, and the verb will always be singular+ 3rd person to generate English equivalent we should use 'being + 3rd form of the verb'.
9	Conjunction Divergence	In Sanskrit the conjunction like ' vaa , athvaa' plays multiple roles in sentence formation. The sentences (i) to (iii) in example shows the divergence ('or' in English).	(i) (a) SS: रवि पटियालाम् गतवान अस्ति वा चन्डीगढम्(Ravi Pattiyaalaam Gatavaana Asti Vaa Canddiigaddham) (b) ES: Ravi has gone either to Patiala or to Chandigarh. (ii) (a) SS: हिन्दीम् वा संस्कृतम् वदतु || हिन्दीम् अथवा संस्कृतम् वदतु (Hindiim Vaa Samskrtam Vadatu || Hindiim Athavaa Samskrtam Vadatu) (b) ES: Speak in Hindi or Sanskrit. (iii) (a) SS: किम सीता गच्छति वा आगच्छति(Kima Siitaa Gacchati Vaa Aagacchati) (b) ES: Does Sita going or coming? In the above sentences 'vaa, athvaa' acts as coordinate conjunction ('either-or || or' in English) in Sanskrit which joins two clauses as in sentence (i) and two phrases as in sentence (ii). Moreover, we have 'vaa and athvaa' in Sanskrit for single 'or' in the English language.	To identify such divergence, we have to see in Sanskrit sentence the occurrence of 'vaa, athvaa' and the solution is by using "Either-or || or" in English equivalent.
10	Word order Divergence	Although the Sanskrit language is a free word order language, i.e., Subject (S), Object (O), Verb (V) could come at any position, but in case of interrogative sentences this free word order characteristic creates a problem as shown in Example 1 sentence.	(i) (a) SS: किम् रामः पठति ? रामः किम् पठति ? रामः पठति किम् ? (Kim Raamah Patthati ? Raamah Kim Patthati ? Raamah Patthati Kim ?) (b) ES: Is Ram studying? What is Ram studying? Is Ram studying?	Reordering of the target sentence as per the English language format.
11	Tense Divergence	A single Sanskrit sentence is realized by two sentences in English which show divergence in both directions as shown in example i	(i) (a) SS: (Sah Vidyaalayam Gacchati) (सः विद्यालयम् गच्छति) (b) ES: He goes to school. (c) ES: He is going to school.	Table B1 of Appendix B is used to recommend a solution for such divergence.

Three parsers for Sanskrit Language have been proposed: shallow parser [67], deterministic parser [68] and constraint-based parser [69]. Bhadra et al. [70] proposed a Sanskrit analysis system which performs segmentation using Sandhi module, shallow parsing and Karaka Analysis module to analyze Sanskrit sentences. For processing Sanskrit compounds, [71] proposed a system which performs the segmentation task automatically and identifies the Sanskrit compounds using statistical techniques. Using Anusaaraka platform [17, 72] proposed a Sanskrit-to-Hindi MTS.

Only two MTS have been reported for Sanskrit-to-English translation: one by [73] and the other by [74]. Aparna proposed an RBMT MTS in 2005 for Sanskrit text to English translation. It performed morphological analysis of the input Sanskrit sentence and applied Sandhi rules in reverse to generate the simple words from the compound Sanskrit words. These simple words were further sent to transducer module where word-by-word transducers were generated. A set of transducers was used for generating the parser, and the output was sent to translator module which generated the target language. Upadhayay et al. proposed another DMT system in 2014 for translating Sanskrit text to English text and also provided text to speech conversion, but the system did not perform any syntactic or semantic analysis. It merely performed word-for-word replacement and reordering of output words.

From the above discussion, it is found that several approaches have been used in MT systems development and specifically for Sanskrit language also like [17, 72, 74] have used DMT approach, while others [73] have used RBMT approach for developing translation for Sanskrit to English language. Each method has its advantages as well as disadvantages. The complexity in designing the rule base to cover big portion of the language has been the challenge for the researchers, and the problem of lack of semantic and syntactic analysis which results in less efficient translation for large sentences in DMT approach

gives motivation to the authors to propose a hybrid form of both DMT and RBMT approaches to make advantages of both the approaches.

3 Language divergence among Sanskrit and English: identification and recommendation

It is necessary to understand the divergence among the languages under consideration before starting the translation process. Dorr [10] classified the language divergence problem into seven categories based on lexical and semantic attributes with their possible solutions. Figure 2 shows the classification of divergence. Goyal and Sinha [11] and Mishra and Mishra [12] also identified the language divergence among English and Sanskrit languages, which includes Dorr's classification and other divergence patterns also. Mishra and Mishra [12] provided the recommended solutions for the divergence identified in the form of algorithms. Three types of information are required to provide a solution to any divergence: GLR, CLR and LCS. GLR and CLR are languages independent, whereas LCS stores the language-dependent information about lexical items. An exception in GLR or CSR or both in either of the languages indicates the occurrence of lexical divergence. Table 1 shows various types of divergence between Sanskrit and English languages with possible recommendations during translation. In Table 1, "SS" denotes Sanskrit sentence and "ES" denotes the English sentence. The Sanskrit sentences are written in both unicode and Indian language TRANSliteration (ITRANS) formats.

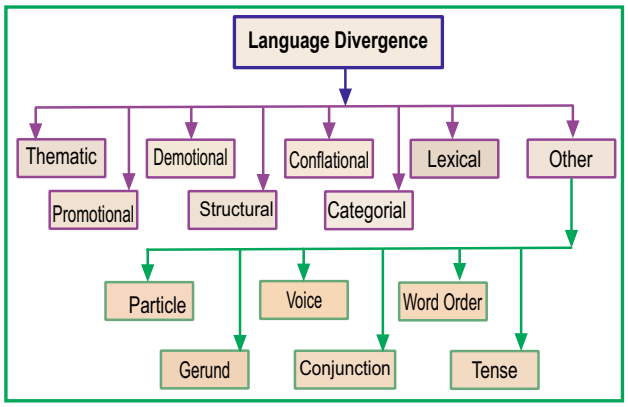

Fig. 2 Language divergence

4 Proposed system

The proposed Sanskrit-to-English translation system uses a hybrid approach of rule-based and direct machine translation technique. The system uses unicode representation for the Sanskrit language sentence. The proposed Sanskrit-to-English MTS is divided into six modules for translation as shown in Fig. 3.

4.1 Module 1

This module performs the preprocessing and part-of-speech tagging of the source language sentence.

4.1.1 Source language preprocessing

Although Sanskrit is a free word order language, the proposed system uses the subject–object–verb (SOV) order for processing Sanskrit sentences as this format is followed by other Indian languages as well. In this phase, the Sanskrit sentence is taken as input using unicode encoding scheme. The input sentence is checked for the SOV grammar structure and restructured into the SOV format if it does not follow the required grammar structure. The reformatting is done using the Kaarka Analysis (Case structure)) Sanskrit grammar rule base, which identifies the subject, object and verb from the input sentence.

The three different forms for the sentence *Ram goes to school* are as follows:

रामः विद्यालयम् गच्छति	रामः गच्छति विद्यालयम्	गच्छति विद्यालयम् रामः
Raamah Vidyaalayam Gacchati	Raamah Gacchati Vidyaalayam	Gacchati Vidyaalayam Raamah
(Ram) (to school) (goes)	(Ram (goes) (to school)	(goes) (to school) (Ram)
S O V	S V O	V O S

In Sanskrit, the best feature is that the position of the word does not tell about the role in the sentence. So the grammatical characteristics (grammar rule base) are used by the system to identify which word plays the role of subject, object or verb and accordingly convert the sentence into SOV format for easy translation. SOV format is used because other Indian languages use the same format for translation. However, there are exceptions to this in the case of interrogative Sanskrit sentence as discussed in the language divergence section earlier. After finalizing the word order, the tokens are generated by using space as the delimiter and forwarded to the next module, and if the word is a compound word, then Sandhi rule base is applied in reverse to get the desired tokens.

4.1.2 Part-of-speech tagger

This is the most important part of the system because the translation of the Sanskrit sentence highly depends on the correct POS tag. Several POS tagsets [75, 76] have been proposed including [77, 78] (JPOS). The comparison of these tagsets is presented in Table 2 based on well-defined criteria.

The URLs for the above-mentioned POS taggers are: https://www.sketchengine.eu/tagset-indian-languages/, http://sanskrit.jnu.ac.in/corpora/JNU-Sanskrit-Tagset.htm, http://www.ldcil.org/standardsTextPOS.aspx, http://sanskrit.jnu.ac.in/corpora/MSRI-JNU-Sanskrit-Tagset.htm and http://sanskrit.jnu.ac.in/cpost/post.jsp, respectively.

Based on the comparison in Table 2, the IL-POSTS Sanskrit tagset has been selected for the proposed translation system . A Sanskrit tagged words dataset of more than four lakhs has been prepared manually. Further, to improve the efficiency of the tagger Sanskrit rule base, Sanskrit–English bilingual dictionary and Sanskrit–UNL dictionary have also been used. The process of POS tagging is as follows:

(a) The tokens generated from the last phase are processed first by rule base.
(b) By applying the rules, the tagging is done token by token and forwarded to next phase.
(c) If no rule is found for any token, then the tagged corpus is used to do the tagging with elasticsearch technique to enhance the processing speed.
(d) If ambiguity persists, then the Sanskrit–UNL dictionary is used to disambiguate the tokens by using UNL attributes and tagging is done accordingly. The tagged tokens are sent to Module 3 for processing.

4.2 Module 2

This module is the database for the proposed system which consists of

(a) A Sanskrit–English bilingual dictionary of more than two lakhs words.
(b) Sanskrit–UNL dictionary of 17,000 words of the general domain [79].
(c) A Sanskrit tagged corpus of more than four lakhs entries.

As shown in Fig. 3 this database is used by various modules in the translation process. For enhancing the data access from these dictionaries and tagged dataset, the authors have used elasticsearch technique which is an open-source, scalable, text search and analytical engine. It performs the task of indexing words and makes them ready to be searched with their position rapidly. It allows the analysis, searching and storing of large volumes of data quickly and in near real time. Searching can be done on any data whether structured or unstructured using this technique.

4.3 Module 3

This module demonstrates the Sanskrit grammar and the CYK parser used for processing input Sanskrit language text. This module is further divided into three submodules: Sanskrit grammar, CYK parsing table and Sanskrit parse tree generation.

4.3.1 Sanskrit grammar

This section describes the Sanskrit grammar. The authors have designed a context-free grammar for the Sanskrit language processing. The designed grammar is as follows:

$G = \{N, \sum, P, S\}$

where N={ S, NP(obj), Predicate, NP(conj)} //set of Non-terminal symbols ,

$\sum = \{$ NP(subj), VP, Conj , NP(Ind_obj)$\}$ //set of Terminal symbols ,

P is the set of production rules.

$$
\begin{aligned}
\text{P=}\quad & S \rightarrow NP(subj)\,Predicate \\
& \quad | \; NP(conj)\,Predicate \\
& NP(obj) \rightarrow NP(obj)\,NP(Ind\_obj) \\
& \quad | NP(Ind\_obj)\;\; NP(obj) \\
& Predicate \rightarrow \; NP(obj)\,VP \\
& NP(conj) \rightarrow NP(subj)\,Conj \\
& \quad | \; NP(subj)\,NP(conj)
\end{aligned}
$$

S=S // start symbol. Since the CYK parser uses only CNF form of the CFG grammar. So the CFG grammar was converted into CNF form as follows:

G_1={ N_1, $\sum_1$, P_1,S }.

Here, N_1={ S, NP(obj), Predicate, NP(conj), V,X,A,B} //set of Non-terminal symbols

$\sum_1$= { NP(subj), VP, Conj, NP(Ind_obj)} //set of Terminal symbols

P is the set of production rules.

$$
\begin{aligned}
\text{P}_1\text{=}\quad & S \rightarrow X \;\; Predicate \;\; | \;\; NP(conj)\,Predicate \\
& NP(obj) \rightarrow \;\; NP(obj)\,A \;\; | \;\; A \;\; NP(obj) \\
& Predicate \rightarrow NP(obj)\,V \\
& NP(conj) \rightarrow X \;\; B \;\; | \;\; X \;\; NP(conj) \\
& X \rightarrow NP(subj) \\
& A \rightarrow NP(Ind\_obj) \\
& V \rightarrow VP \\
& B \rightarrow Conj
\end{aligned}
$$

S=S // start symbol.

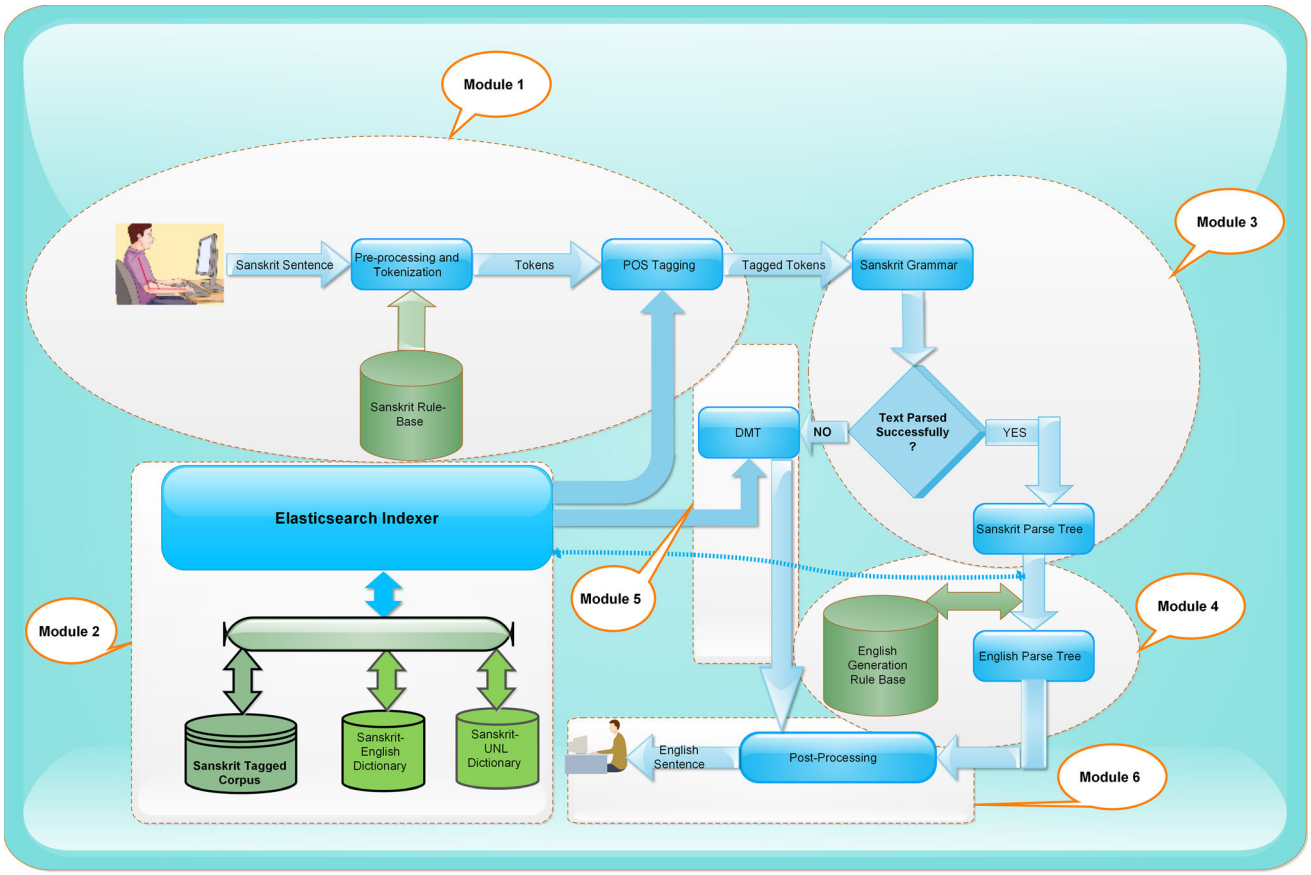

Fig. 3 Architecture of the proposed Sanskrit-to-English MT system

4.3.2 CYK parsing table

The grammar developed in the previous section is implemented using CYK Parser, which uses bottom-up parsing and dynamic programming approach. The CYK parser works on CFG grammar in CNF form. For any input string of length "m" and the grammar with "p" non-terminals, it starts in a triangular form. The worst-case time complexity of the CYK parser is $O(m^3)$ [80] and space complexity is $O(m^2)$ [81], which are better than other parsing algorithms in worst-case scenario, where m represents the input string length (number of words in the input Sanskrit sentence). It makes use of matrix representation for performing the parsing task. The process of parsing is as follows:

(a) Take words and their part of speech as input.
(b) Create a matrix of size $[N, N]$ where N is the number of tokens in the sentence.
(c) Fill the diagonal cells of the matrix with the mapped grammar's variables and terminals in the same order as the tokens are present in the sentence.
(d) Write variable present on the left side in any production rule of the grammar at position $[i, j]$, if the right side of that production could be broken down into two parts. The first part is present at $[i, x]$ where x>i and x<j and second part is present at $[y, j]$ where $y < j$ and $y > i$.
(e) Whenever there is more than one possibility, CYK implementation considers the one which is discovered at the later stage (the last one overwrites all previous reduction decisions in the case of any overlapping).
(f) Convert the CYK matrix into an actual tree by beginning from start symbol of grammar present at $[0, N]$ and tracing children at each point.

Table 4 of "Appendix 1" shows step-by-step generation of parsing table for the input Sanskrit sentence विशालः ओदनम् चमसेन मयंकस्य थालिकायाः खादति Vishaalah Odanam Camasena Mayamkasya Thaalikaayaah Khaadati) using CNF grammar production.

If the input sentence is processed successfully by the proposed grammar, then the parse table is used to generate the Sanskrit parse tree in Sect. 4.3.3, and if not then, the control goes to Sect. 4.5.

Table 2 POS tagset comparison

	IIT/ILMT	JPOS	LDC-IL	IL-POSTS	CPOS
Common/Sanskrit	Common	Sanskrit	Common	Common	Sanskrit
Fine/coarse grained	Coarse	Fine	Fine	Fine	Coarse
Flat/hierarchy	Flat	Flat	Flat	Hierarchy	Flat
Base	Penn Tree Bank	Paninian grammar	ILMT	EAGLES	ILMT + JPOS
Multilingual support	Yes	No	Yes	Yes	No
Number of tags	26	134	26	7 (cat)+ 11 (attributes)	28

4.3.3 Sanskrit tree generation

Proposed Algorithm 1 is used to generate the parse tree from the parsing table generated in Sect. 4.3.2. Figure 6 of "Appendix 1" shows the example of Sanskrit parse tree generated from the parsing Table 4. The semantic information is obtained from Sanskrit–UNL dictionary [79].

4.4 Module 4

This module generates the English parse tree equivalent to Sanskrit parse tree. It uses the English generation rule base as shown in Table 5 of "Appendix 2," language divergence rule as discussed in Sect. 3 and the bilingual dictionary (Sanskrit–English) for generating the target parse tree. Elasticsearch technique reduces the delay in accessing the data from bilingual dictionaries. Sanskrit–UNL dictionary removes the ambiguity among words during target language generation and adds the semantic information to the target sentence. Figure 7 of "Appendix 1" shows the example of English parse tree generated from the equivalent Sanskrit parse tree.

4.5 Module 5

This module performs the translation using DMT approach. The bilingual Sanskrit–Hindi dictionary and Sanskrit–UNL dictionary are used to generate the target language word for the source language word. The word-by-word replacement is done in this phase. Due to Module 2, the processing speed of accessing the equivalent English word is enhanced. The reordering of word as per target language is done in Sect. 4.6.

4.6 Module 6

In this phase, the final output is generated and reordering is performed to get the final output. In module 4, scanning the leaf nodes of the tree from left to right generates the target English sentence. The word order of the Sanskrit language is subject–object–verb, whereas for English it is subject–verb–object, so in the final English tree the verb part and the object part are mutually shifted to the desired output. In module 5 for direct approach, the reordering of words is done to get the sentence in target language structure.

5 Experimental setup

The two bilingual dictionaries and a tagged corpus for implementing the proposed system were developed.

(i) Sanskrit–UNL dictionary [79]
This dictionary consists of 17000 Sanskrit–UNL words with grammatical and semantic attributes to remove ambiguity among the words of the sentence.

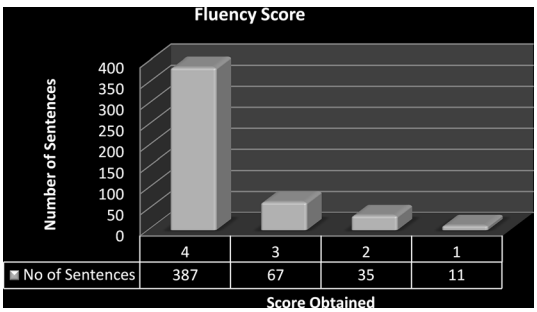

Fig. 4 Fluency score

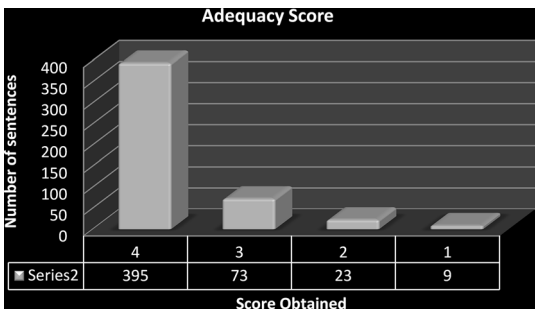

Fig. 5 Adequacy score

(ii) Sanskrit–English Dictionary

This dictionary consists of around 2.5 lakhs Sanskrit–English words and is prepared from the Wikipedia by the authors for the implementation purpose and direct translation.

(iii) Tagged Corpus

A Sanskrit tagged corpus was prepared by combining data from JNU [77] and Gerard Huet, which contains more than 04 lakhs entries. It is used for the POS tagging purpose.

Elasticsearch technique was utilized in module 2 for enhancing the translation speed. The proposed system uses language divergence rule and the Sanskrit–English generation rule base in Table 5 of "Appendix 2." PHP with Xampp server is used to create the Web environment.

Figure 8 of "Appendix 3" shows the processing of the input Sanskrit sentence by tokenizing the sentence and assigning different part-of-speech tags based on knowledge obtained from the Sanskrit rule base/tagged corpus. The tagged tokens are forwarded to the next phase of Sanskrit parsing using Sanskrit CNF grammar with CYK bottom-up parsing algorithm. The CYK parser generates the parsing table. The proposed algorithm 1 which gives the array of left, parent and right nodes of the tree generates the parse tree as shown in Fig. 9 of "Appendix 3." Figure 10 of

Algorithm 1: ParseTree Generation from Parsing Table

Input: Matrix M of order n*n , where n is the number of words in the input sentence
Output: Node list with Left, Parent and Right nodes

for $(i \leftarrow 0$ ***to*** $n-1)$ **do**
  Write Principle diagonal elements of the matrix M as leaf nodes. $Leaf[i] \leftarrow M(i,i)$
  $i \leftarrow i+1$
Take root variable to indicate the root of the tree.
Take three 1-D arrays L, P and R of size n-1 for storing Left, Parent and Right child of the tree.
Take a temporary variable temp and initialize it with value true.
Initialize m to 0;
$m \leftarrow 0$
$temp \leftarrow true$
for $(i \leftarrow 0$ ***to*** $n-2)$ **do**
  $L[m] \leftarrow Leaf[i]$
  for $(j \leftarrow i+1$ ***to*** $n-1)$ **do**
    if $(M(i,j) \neq NULL \wedge temp = true)$ **then**
      $//CellM(i,j)isnotempty$
      make $M(i,j)$ as parent node of $L[m]$
      $P[m] \leftarrow M(i,j)$
      if *(P[m]='S')* **then**
        root=P[m]
      if *(j=i+1)* **then**
        Make $Leaf[j]$ as the right node of the tree
        $R[m] \leftarrow Leaf[j]$
        $m \leftarrow m+1$
      else
        Make $M(i+1,j)$ as the right node of the tree
        $R[m] \leftarrow M(i+1,j)$
        $m \leftarrow m+1$
      $temp \leftarrow false$
    else
      if $(M(i,j) \neq NULL) \wedge (temp = false)$ **then**
        $//Cell$ M(i,j) is not empty
        $L[m] \leftarrow P[m-1]$
        $P[m] \leftarrow M(i,j)$
        if *(P[m]='S')* **then**
          root=P[m]
        $R[m] \leftarrow Leaf[j]$
        $m \leftarrow m+1$
    $j \leftarrow j+1$
  $i \leftarrow i+1$
  $temp \leftarrow true$
for $(j \leftarrow 0$ ***to*** $n-2)$ **do**
  return (L,P,R)

"Appendix 3" shows the generation of English parse tree by using a bilingual dictionary, Sanskrit–UNL dictionary and TLG rule base. The post-processing phase which performs reordering of different phases gives the target language text. If the input sentence is not feasible to parse by the proposed grammar, then the DMT approach is used to do translation with the help of bilingual dictionaries.

6 Results and discussion

A set of manually translated 500 Sanskrit–English sentences were used for the evaluation purpose from the general domain (small stories).

(i) Bilingual Evaluation Understudy (BLEU) [82]
A weighted BLEU score evaluation method is used for evaluating the proposed system. A 2-g cumulative BLEU score is calculated using Natural Language ToolKit (NLTK) in Python language with 0.50 weight for each 1-g, 2-g score, because in 3-g and 4-g model the BLEU score value decreases rapidly as compared to 2-g model. BLEU-2 (2-g BLEU) score of the proposed system is 0.7606.

(ii) Fluency Score [83, 84] The proposed system is also evaluated on a four-scale Fluency score system and achieves a score of 3.63 (out of four). The score indicates the degree with which the generated sentence (by the proposed system) obeys the target language grammar rules. A score of 4 indicates the perfect translation, score of 3 indicates fair translation, score of 2 indicates the acceptable translation and score of 1 indicates incomplete (nonsense) translation.

The analysis of the result of 500 sentences is presented in Fig. 4 and explained as follows:

(a) 387 sentences achieved score 4 (perfect translation)
(b) 67 sentences achieved score 3 (fair translation)
(c) 35 sentences achieved score 2 (acceptable but require efforts to understand)
(d) 11 sentences achieved score 1 (not acceptable)

The above discussion shows that the proposed system generates 90.8 (Score 4 and Score 3 sentences) percent of grammatically correct sentences.

(iii) Adequacy Score [84] indicates the degree with which the information contained in the source language is transferred into the translated sentence. The proposed system obtained 3.72 scores on a four-scale system. A score of 4 indicates the complete information transmission, score of 3 indicates almost complete transmission, score of 2 indicates small information transmission and score

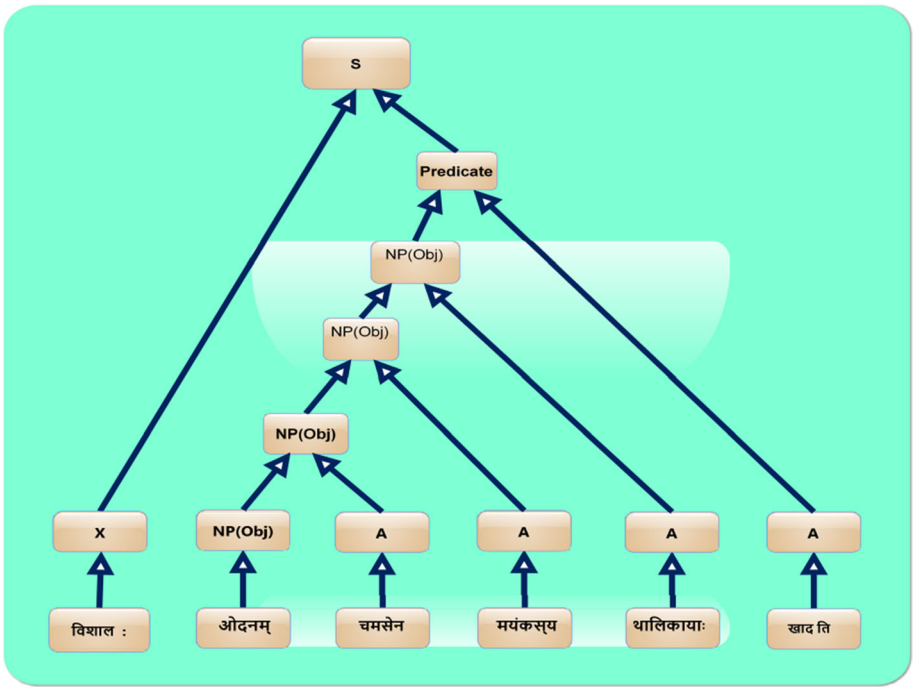

Fig. 6 Sanskrit parse tree

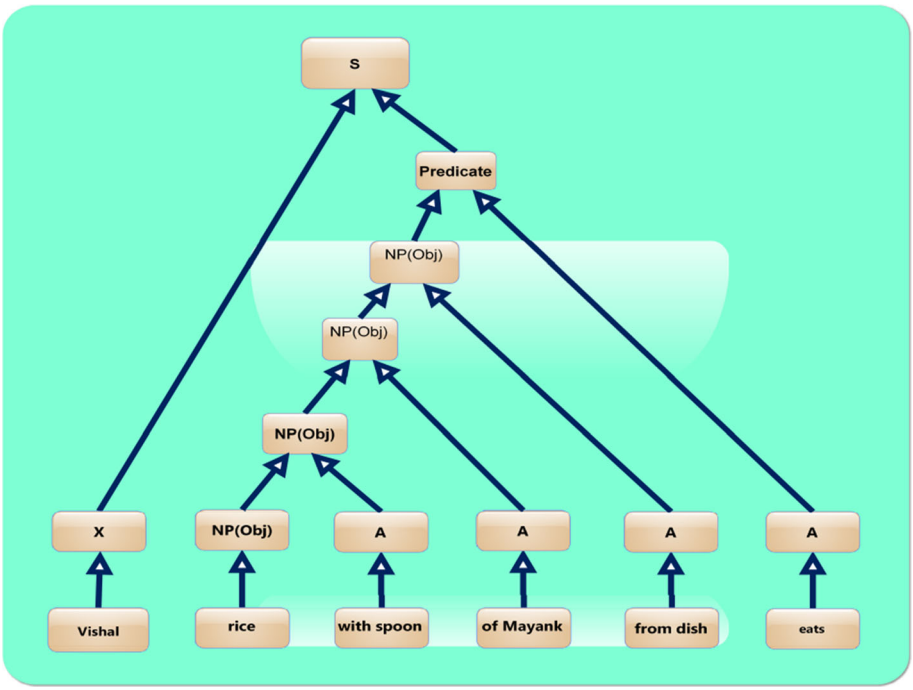

Fig. 7 English parse tree

of 1 indicates no information transmission. The analysis of the result of the 500 sentences is presented in Fig. 5 and explained as follows:

(a) 395 sentences achieved score 4 (complete information transmission)
(b) 73 sentences achieved score 3 (almost complete information transmission)
(c) 23 sentences achieved score 2 (small information transmission)
(d) 09 sentences achieved score 1 (no information transmission)

The above discussion shows that the proposed system transfers 93.6% of source information (Score 4 and Score 3 sentences) successfully in the generated sentences.

Table 3 shows the comparison of the proposed system with other existing systems, and it is found that the proposed system gives better performance in comparison with other existing MT systems. The overall efficiency of the proposed system is 97.8%

7 Conclusion

The proposed Sanskrit-to-English translation system has adopted a hybrid approach of direct machine translation (DMT) and rule-Based machine translation (RBMT) and used POS tagger, Sanskrit grammar, CYK parser and a parsing algorithm. The proposed system has achieved a BLEU score of 0.7606, fluency score of 3.63 and adequacy score of 3.72. The overall efficiency of the developed system is 97.8%.

Proposed POS tagger and parse tree generating algorithm could be used as a Sanskrit language analyzer in developing other NLP applications. Domain-specific

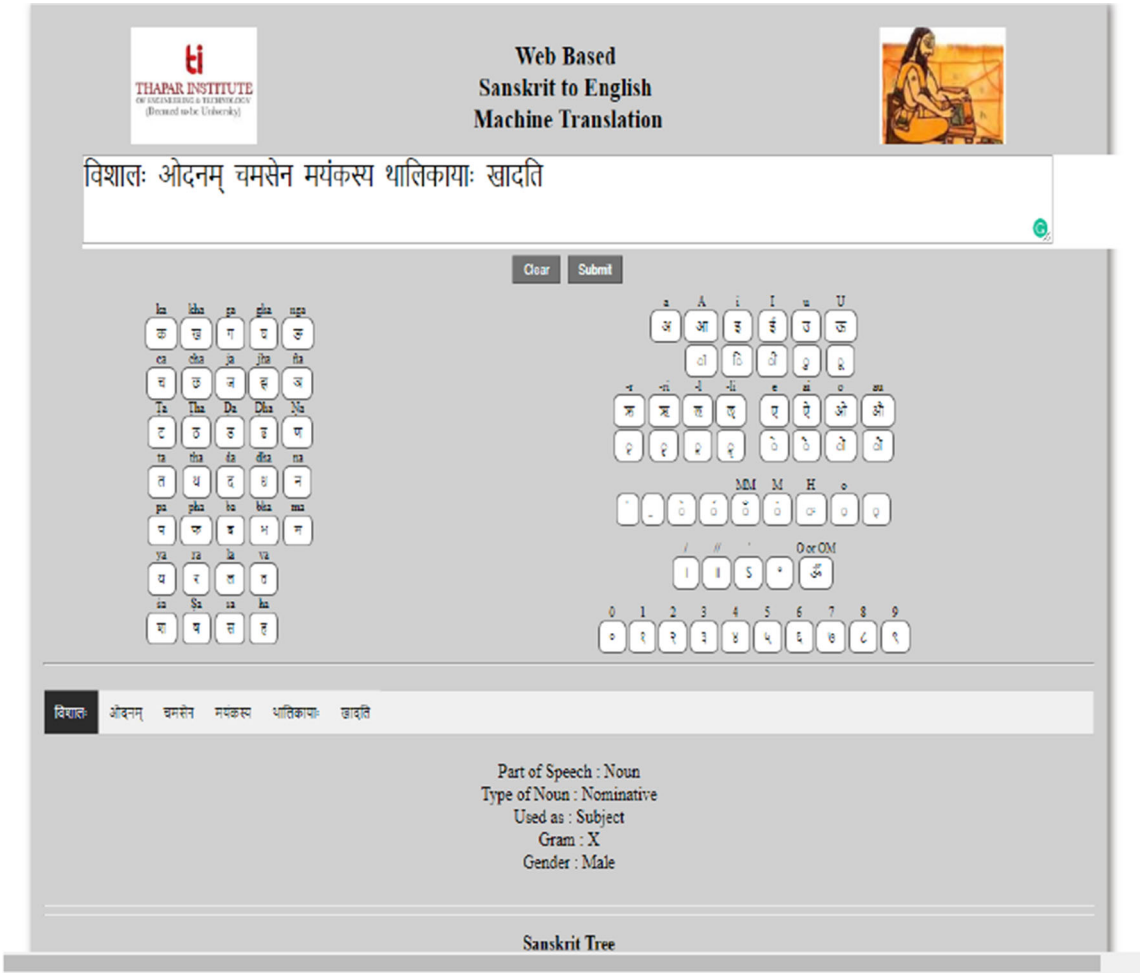

Fig. 8 POS tagging of Sanskrit sentence

applications might be developed by enhancing the particular domain rule base.

Compliance with ethical standards

Conflict of interest We have no conflicts of interest to disclose.

Human and animal rights This article does not contain any studies with animals performed by any of the authors. This article does not contain any studies with human participants or animals performed by any of the authors.

Appendix 1: Generating parsing table and parse tree using CYK parser

In this section, Table 4 explains the processing of Sanskrit text by CYK parser by taking विशालः ओदनम् चमसेन मयंकस्य थालिकायाः खादति as example. Figures 6 and 7 depict the process of parse tree generation from the parsing table.

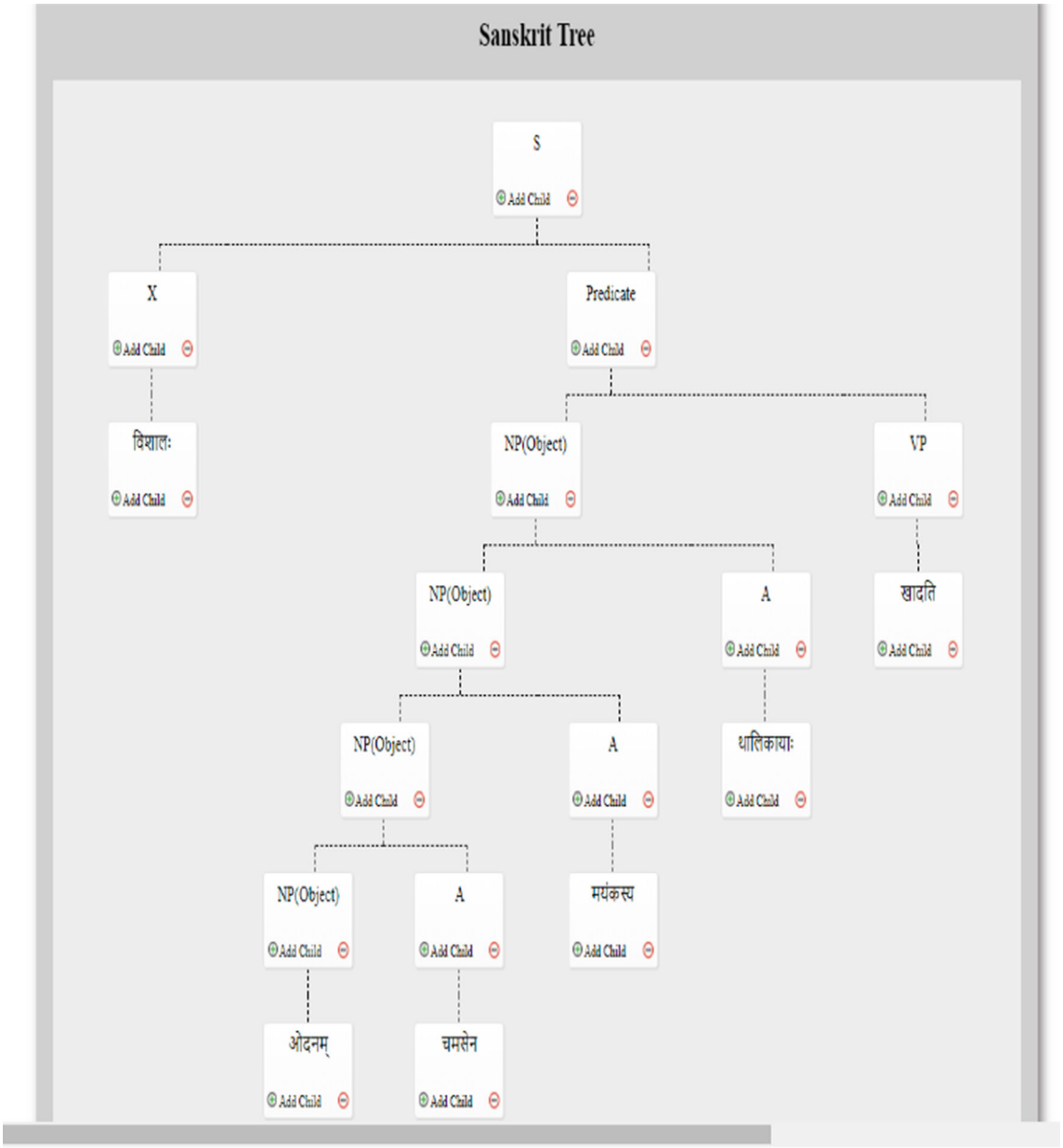

Fig. 9 Sanskrit parse tree

Appendix 2: Target language generation rule base

This section provides the TLGR and covers three voices of Sanskrit language with corresponding English language equivalent. Table 5 shows tabular representation of three voices and ten tenses of Sanskrit with rules to generate English-equivalent translation.

Appendix 3: Implementation of the proposed Sanskrit-to-English MTS

This section shows the software implementation of the proposed Sanskrit-to-English translator using an example.

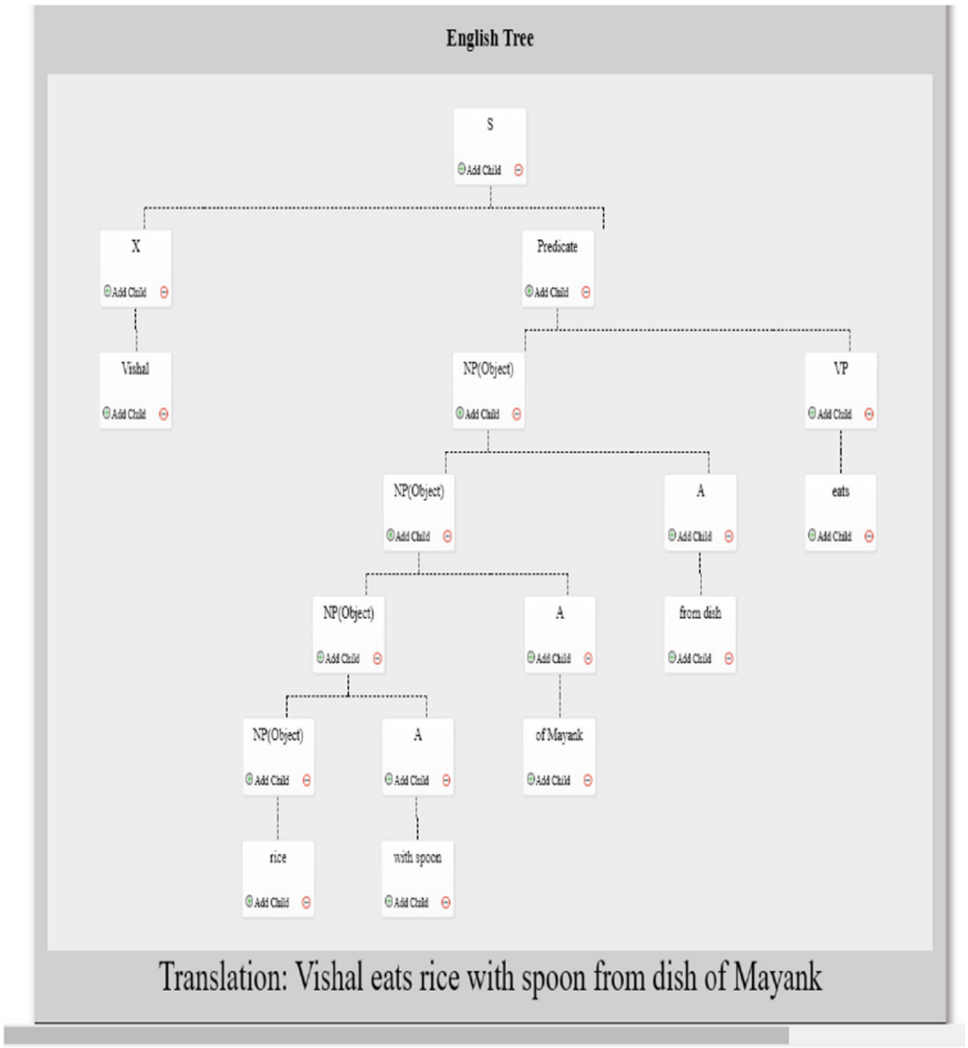

Fig. 10 English parse tree

Table 3 Comparison of the proposed system with other existing systems

MT system	BLEU score	Approach used	Citation
Hindi–English	0.7502	QNN with RBMT	[59]
Sanskrit–English	NA	RBMT	[74]
Hindi–English	0.2182	CBMT	[46]
Sanskrit–English	0.7606	DMT with RBMT	Proposed system

Table 4 CYK processing of Sanskrit text

(a) Processing variables for all length 1 substring

	0	1	2	3	4	5
0	X					
1		NP(Obj)				
2			A			
3				A		
4					A	
5						VP

(b) Processing variables for all length 2 substring

	0	1	2	3	4	5
0	X	-				
1		NP(Obj)	NP(Obj)			
2			A	-		
3				A	-	
4					A	
5						VP

(c) Processing variables for all length 3 substring

	0	1	2	3	4	5
0	X					
1		NP(Obj)	NP(Obj)	NP(Obj)		
2			A	-	-	-
3				A	-	-
4					A	-
5						VP

(d) Processing variables for all length 4 substring

	0	1	2	3	4	5
0	X					
1		NP(Obj)	NP(Obj)	NP(Obj)	NP(Obj)	
2			A	-	-	-
3				A	-	-
4					A	-
5						VP

(e) Processing variables for all length 5 substring

	0	1	2	3	4	5
0	X	-	-	-	-	
1		NP(Obj)	NP(Obj)	NP(Obj)	NP(Obj)	Predicate
2			A	-	-	-
3				A	-	-
4					A	-
5						VP

(f) Processing variables for all length 6 substring

	0	1	2	3	4	5
0	X	-	-	-	-	S
1		NP(Obj)	NP(Obj)	NP(Obj)	NP(Obj)	Predicate
2			A	-	-	-
3				A	-	-
4					A	-
5						VP

(g) Processing of the input Sanskrit sentence

	1	2	3	4	5	6
1	**X** (विशालः)	-(विशालः ओदनम्)	-(विशालः ओदनम् चमसेन)	-(विशालः ओदनम् चमसेन मयंकस्य)	-(विशालः ओदनम् चमसेन मयंकस्य थालिकायाः)	**S** (विशालः ओदनम् चमसेन मयंकस्य थालिकायाः खादति)
2		**NP(Obj)** (ओदनम्)	**NP(Obj)** (ओदनम् चमसेन))	**NP(Obj)** (ओदनम् चमसेन मयंकस्य)	**NP(Obj)**(ओदनम् चमसेन मयंकस्य थालिकायाः)	Predicate (ओदनम् चमसेन मयंकस्य थालिकायाः खादति)
3			**A**(चमसेन)	(चमसेन मयंकस्य)	(चमसेन मयंकस्य थालिकायाः)	(चमसेन मयंकस्य थालिकायाः खादति)
4				**A**(मयंकस्य)	(मयंकस्य थालिकायाः)	(मयंकस्य थालिकायाः खादति)
5					**A**(थालिकायाः)	(थालिकायाः खादति)
6						**VP**(खादति)

Table 5 Target language generation rule base

लट् लकार (Present Tense)			
Active Voice (Subject will be in Nominative Case and Object will be in Accusative Case)			
Person (Subject)	Number (Subject)	Sanskrit Verb	English Verb Form
First ((उत्तम पुरुष)	Singular	Agrees in Person and Number with Subject	V_1 (Present Indefinite)or am + V_1 + ing (Present Continuous)
	Dual/ Plural		are + V_1 +ing (Present Continuous)
Second (मध्यम पुरुष)	Singular/ Dual/ Plural		V_1 (Present Indefinite) or are + V_1 +ing (Present Continuous)
Third (प्रथम पुरुष)	Singular		s/es + V_1 (Present Indefinite) or is + V_1 +ing (Present Continuous)
	Dual/ Plural		V_1 (Present Indefinite) or are + V_1 +ing (Present Continuous)
Passive Voice(Object will be in Nominative Case and Subject will be in instrumental Case)			
Person (Object)	Number (Object)	Sanskrit Verb	English Verb Form
First ((उत्तम पुरुष)/ Second (मध्यम पुरुष)/ Third (प्रथम पुरुष)	Singular	Agrees in Person and Number with Subject	Is+ V_3 (Passive form of Present Indefinite)
	Dual/ Plural		are+ V_3(Passive form of Present Indefinite)
Abstract Voice (Subject and Object both will be in Instrumental Case)			
Person (Object)	Number (Object)	Sanskrit Verb	English Verb Form
First ((उत्तम पुरुष)/ Second (मध्यम पुरुष)	Singular (Object)	Verb will be in singular and 3rd person form.	Is+ being+ V_3 (Passive form of Present Continuous)
Third (प्रथम पुरुष)	Dual/ Plural (Object)		are+being+ V_3 (Passive form of Present Continuous)
लङ् लिट् and लुङ् लकार (Past Tense)			
Active Voice (Subject will be in Nominative Case and Object will be in Accusative Case)			
Person (Subject)	Number (Subject)	Sanskrit Verb	English Verb Form
First ((उत्तम पुरुष)	Singular	Agrees in Person and Number with Subject	V_2 (Past Indefinite) or was + V_1 +ing (Past Continuous)
	Dual/ Plural		were + V_1 +ing (Past Continuous)
Second (मध्यम पुरुष)	Singular/ Dual/ Plural		V_2 (Past Indefinite) or were + V_1 +ing (Past Continuous)
Third (प्रथम पुरुष)	Singular		V_2 (Past Indefinite) or was + V_1 +ing (Past Continuous)
	Dual/ Plural		were + V_1 +ing (Past Continuous)
Passive Voice(Object will be in Nominative Case and Subject will be in instrumental Case)			
Person (Object)	Number (Object)	Sanskrit Verb	English Verb Form
First ((उत्तम पुरुष) / Second (मध्यम पुरुष)/ Third (प्रथम पुरुष)	Singular	Agrees in Person and Number with Subject	was+ V_3 (Passive form of Past Indefinite)
	Dual/ Plural		were+ V_3(Passive form of Past Indefinite)
Abstract Voice (Subject and Object both will be in Instrumental Case)			
Person (Object)	Number (Object)	Sanskrit Verb	English Verb Form
First ((उत्तम पुरुष)/ Second (मध्यम पुरुष)/ Third (प्रथम पुरुष)	Singular	Verb will be in singular and 3rd person form only.	was+ being+ V_3 (Passive form of Past Continuous)
	Dual/ Plural		were+being+ V_3 (Passive form of Past Continuous)
लिट् लकार (Past Perfect)			
Active Voice (Subject will be in Nominative Case and Object will be in Accusative Case)			
Person (Subject)	Number (Subject)	Sanskrit Verb	English Verb Form
First ((उत्तम पुरुष)/ Second (मध्यम पुरुष)/ Third (प्रथम पुरुष)	Singular / Dual/ Plural	Agrees in Person and Number with Subject	had+V_3 (Past Perfect)
लुङ् लकार (Aorist/ Past Perfect Continuous)			
Active Voice (Subject will be in Nominative Case and Object will be in Accusative Case)			
Person (Subject)	Number (Subject)	Sanskrit Verb	English Verb Form
First ((उत्तम पुरुष)/ Second (मध्यम पुरुष)/ Third (प्रथम पुरुष)	Singular / Dual/ Plural	Agrees in Person and Number with Subject	Had+been+V_3
लृट् लकार (Simple Future)			
Active Voice (Subject will be in Nominative Case and Object will be in Accusative Case)			
Person (Subject)	Number (Subject)	Sanskrit Verb	English Verb Form
First ((उत्तम पुरुष)	Singular / Dual/ Plural	Agrees in Person and Number with Subject	will/shall+ V_1
Second (मध्यम पुरुष)/ Third (प्रथम पुरुष)	Singular / Dual/ Plural		will + V_1
लृट् लकार (Future Continuous)			

Table 5 continued

Active Voice (Subject will bein Nominative Case and Object will be in Accusative Case)			
Person (Subject)	Number (Subject)	Sanskrit Verb	English Verb Form
First ((उत्तम पुरुष)	Singular / Dual/ Plural	Agrees in Person and Number with Subject	will/shall + be+ V_1+ing
Second (मध्यम पुरुष)/ Third (प्रथम पुरुष)	Singular / Dual/ Plural		will + be+ V_1+ing
लोट् लकार (Imperative Mood)			
Person (Subject)	Number (Subject)	Sanskrit Verb	English Verb Form
First ((उत्तम पुरुष)/ Second (मध्यम पुरुष)/ Third (प्रथम पुरुष)	Singular / Dual/ Plural	Agrees in Person and Number with Subject	Let+V_1 +! must+V_1 +! (In case of Command) Or Please + V_1(In case of Request) Or Can+ V_1 + ? (in case of question)
विधिलिङ् लकार (Potential Mood)			
Person (Subject)	Number (Subject)	Sanskrit Verb	English Verb Form
First ((उत्तम पुरुष)/ Second (मध्यम पुरुष)/ Third (प्रथम पुरुष)	Singular / Dual/ Plural	Agrees in Person and Number with Subject	May+V_1(Possibility) Should+V_1(Advice) Should+be+V_3(Appropriatness) Should+have+V_3 (possibility) Should+not+ V_1(Notice)
आशिलिङ् लकार(Benedictive Mood)			
Person (Subject)	Number (Subject)	Sanskrit Verb	English Verb Form
First ((उत्तम पुरुष)/ Second (मध्यम पुरुष)/ Third (प्रथम पुरुष)	Singular / Dual/ Plural	Agrees in Person and Number with Subject	May+V_1(blessing)
लृङ् लकार (Conditional Mood)			
Person (Subject)	Number (Subject)	Sanskrit Verb	English Verb Form
First ((उत्तम पुरुष)/ Second (मध्यम पुरुष)/ Third (प्रथम पुरुष)	Singular / Dual/ Plural	Agrees in Person and Number with Subject	If+V_1

References

1. Kak SC (1987) The paninian approach to natural language processing. Int J Approx Reason 1(1):117–130
2. Briggs R (1985) Knowledge representation in Sanskrit and artificial intelligence. AI Mag 6(1):32
3. Bahadur P, Jain A, Chauhan DS (2011) English to Sanskrit machine translation. In: Proceedings of the international conference & workshop on emerging trends in technology. ACM, pp 641–645
4. Mishra V, Mishra RB (2008) Study of example based English to Sanskrit machine translation. J Res Dev Comput Sci Eng 37:43–54
5. Mishra V, Mishra RB (2009) Ann and rule based model for English to Sanskrit machine translation. INFOCOMP J Comput Sci 9(1):80–89
6. Bahadur P, Jain AK, Chauhan DS (2012) 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, pp 52–59
7. Lewis MP, Simons GF, Fennig CD (2015) Ethnologue: languages of Ecuador. SIL International, Dallas
8. Mallikarjun B (2010) Patterns of Indian multilingualism. In: Strength for today and bright hope for tomorrow, vol 10, no 6, pp 1–18
9. Dorr BJ , Hovy EH, Levin LS (2004) Natural language processing and machine translation encyclopedia of language and linguistics, (ELL2). Machine translation: interlingual methods. In: Proceeding international conference of the world congress on engineering
10. Dorr Bonnie J (1994) Machine translation divergences: a formal description and proposed solution. Comput Linguist 20(4):597–633
11. Goyal P, Sinha RMK (2009) Translation divergence in English–Sanskrit–Hindi language pairs. In: International sanskrit computational linguistics symposium. Springer, pp 134–143
12. Mishra V, Mishra RB (2009) Divergence patterns between English and Sanskrit machine translation. INFOCOMP 8(3):62–71
13. Goyal V, Lehal GS (2010) Web based Hindi to Punjabi machine translation system. J Emerg Technol Web Intell 2(2):148–151
14. Dubey P et al (2013) Machine translation system for Hindi–Dogri language pair. In: 2013 international conference on machine intelligence and research advancement (ICMIRA). IEEE, pp 422–425
15. Dubey P (2019) The Hindi to Dogri machine translation system: grammatical perspective. Int J Inf Technol 11(1):171–182
16. Narayana VN (1994) Anusarak: a device to overcome the language barrier. PhD thesis, Ph.D. thesis, Department of CsE, IIT Kanpur
17. Bharati A, Chaitanya V, Kulkarni AP, Sangal R (1997) Anusaaraka machine translation in stages. VIVEK-Bombay 10:22–25
18. Bharati RM, Sankar B, Reddy P, Sharma DM, Sangal R (2003) Machine translation: the shakti approach. Pre-conference tutorial. In: ICON
19. Josan GS, Lehal GS (2008) A Punjabi to Hindi machine translation system. In: 22nd international conference on on

computational linguistics: demonstration papers. Association for Computational Linguistics, pp 157–160
20. Rajan R, Sivan R, Ravindran R, Soman KP (2009) Rule based machine translation from English to Malayalam. In: ACT'09. International conference on advances in computing, control, & telecommunication technologies, 2009. IEEE, pp 439–441
21. Goyal P, Sinha RMK (2009) A study towards design of an English to Sanskrit machine translation system. In: Sanskrit computational linguistics. Springer, pp 287–305
22. Pathak GR, Godse SP (2010) English to Sanskrit machine translation using transfer approach. In: International conference on methods and models in science and technology. American Institute of Physics, Pune, pp 122–126
23. Mishra V, Mishra RB (2012) English to Sanskrit machine translation system: a rule-based approach. Int J Adv Intell Paradig 4(2):168–184
24. Reddy MV, Hanumanthappa M (2013) Indic language machine translation tool: English to Kannada/Telugu. In: Multimedia processing, communication and computing applications. Springer, New Delhi, pp 35–49. https://doi.org/10.1007/978-81-322-1143-3_4
25. Jayan V, Bhadran VK (2014) Anglabharati to Anglamalayalam: an experience with English to Indian language machine translation. In: 2014 international conference on contemporary computing and informatics (IC3I). IEEE, pp 282–287
26. Desai P, Sangodkar A, Damani OP (2014) A domain-restricted, rule based, English–Hindi machine translation system based on dependency parsing. In: Proceedings of the 11th international conference on natural language processing, pp 177–185
27. Balyan R, Chatterjee N (2015) Translating noun compounds using semantic relations. Comput Speech Lang 32(1):91–108
28. Aasha VC, Ganesh A (2015) Machine translation from English to Malayalam using transfer approach. In: 2015 international conference on advances in computing, communications and informatics (ICACCI). IEEE, pp 1565–1570
29. Sridhar R, Sethuraman P, Krishnakumar K (2016) English to Tamil machine translation system using universal networking language. Sādhanā 41(6):607–620
30. Sinha R, sivaraman KS, Agrawal A, Jain R, Srivastava R, Jain A et al (1995) Anglabharti: a multilingual machine aided translation project on translation from English to Indian languages. In: IEEE international conference on systems, man and cybernetics, 1995. Intelligent systems for the 21st century, vol 2. IEEE, pp 1609–1614
31. Darbari H (1999) Computer-assisted translation system—an Indian perspective. In: Machine translation summit VII, 13th–17th September, pp 80–85
32. Dave S, Parikh J, Bhattacharyya P (2001) Interlingua-based English-Hindi machine translation and language divergence. Mach Transl 16(4):251–304
33. Singh S, Dalal M, Vachani V, Bhattacharyya P, Damani OP (2007) Hindi generation from interlingua. In: Proceedings of machine translation summit, pp 1–8
34. Choudhary A, Singh M (2009) Gb theory based Hindi to English translation system. In: 2nd IEEE international conference on computer science and information technology, 2009. ICCSIT 2009. IEEE, pp 293–297
35. Christopher M, Rao UM (2010) IL-ILMT sampark: a hybrid machine translation system. In 32nd all India conference of linguistics (AICL32). Lucknow University, Lucknow, pp 69–75
36. Batra KK, Lehal GS (2010) Rule based machine translation of noun phrases from Punjabi to English. Int J Comput Sci Issues 7(5):409–413
37. Batra KK, Lehal GS (2011) Automatic translation system from Punjabi to English for simple sentences in legal domain. Int J Trans 23(1):79–98
38. Kumar P, Sharma RK (2012) Punjabi to unl enconversion system. Sadhana 37(2):299–318
39. Parteek Kumar and Rajendra Kumar Sharma (2013) Punjabi deconverter for generating Punjabi from universal networking language. J Zhejiang Univ Sci C 14(3):179–196
40. Udupa UR, Faruquie TA (2005) An English–Hindi statistical machine translation system. In: Su KY, Tsujii J, Lee JH, Kwong OY (eds) Natural language processing–IJCNLP 2004. IJCNLP 2004. Lecture notes in computer science, vol 3248. Springer, Berlin, Heidelberg, pp 254–262. https://doi.org/10.1007/978-3-540-30211-7_27
41. Antony PJ (2013) Machine translation approaches and survey for Indian languages. Int J Comput Linguist Chin Lang Process 18(1):47–78
42. Garje GV, Kharate GK (2013) Survey of machine translation systems in India. Int J Nat Lang Comput (IJNLC) 2(4):47–67
43. Sinha RMK (2004) An engineering perspective of machine translation: anglabharti-ii and anubharti-ii architectures. In: Proceedings of international symposium on machine translation, NLP and translation support system (iSTRANS-2004), pp 10–17
44. Jain R Sinha RMK, Jain A (2001) Anubharti-using hybrid example-based approach for machine translation. In: STRANS-2001, IIT Kanpur, pp 20–32
45. Sinha RMK, Thakur A (2005) Machine translation of bi-lingual Hindi–English (Hinglish) text. In: 10th Machine translation summit (MT Summit X), Phuket, Thailand, pp 149–156
46. Sachdeva K, Srivastava R, Jain S, Sharma DM (2014) Hindi to English machine translation: using effective selection in multi-model SMT. In: LREC, pp 1807–1811
47. Dungarwal P, Chatterjee R, Mishra A, Kunchukuttan A, Shah R, Bhattacharyya P (2014) The IIT bombay Hindi–English translation system at WMT 2014. In: ACL 2014, p 90
48. Och FJ (2007) Google translator. In: Joint conference on empirical methods in natural language processing and computational natural language learning. Prague. Association for Computational Linguistics, pp 858–867
49. Venkatapathy S, Bangalore S (2009) Discriminative machine translation using global lexical selection. ACM Trans Asian Lang Inf Process (TALIP) 8(2):8
50. Sharma N (2011) English to Hindi statistical machine translation system. PhD thesis, Thapar University Patiala
51. Khan N, Anwar W, Bajwa UI, Durrani N (2013) English to Urdu hierarchical phrase-based statistical machine translation. In: WSSANLP2013, Japan, October 2013, pp 72–76
52. Ali A, Hussain A, Malik MK (2013) Model for English–Urdu statistical machine translation. World Appl Sci 24:1362–1367
53. Sheikh M, Conlon S (2013) Application of machine translation in bilingual knowledge management. Int J Intercult Inf Manag 3(2):123–137
54. Jawaid B, Kamran A, Bojar O (2014) English to Urdu statistical machine translation: establishing a baseline. In: Proceedings of the Fifth workshop on south and southeast Asian natural language processing, pp 37–42
55. Naskar S, Bandyopadhyay S (2005) Use of machine translation in India: current status. AAMT J 16:25–31
56. Badodekar S (2003) Translation resources, services and tools for Indian languages. In: Computer science and engineering department, Indian Institute of Technology, Mumbai, 400019
57. Saini TS, Lehal GS, Kalra VS (2008) Shahmukhi to Gurmukhi transliteration system. In: 22nd international conference on on computational linguistics: demonstration papers. Association for Computational Linguistics, pp 177–180
58. Goyal V, Lehal GS (2011) Hindi to Punjabi machine translation system. In: Proceedings of the 49th annual meeting of the association for computational linguistics: human language

technologies: systems demonstrations. Association for Computational Linguistics, pp 1–6
59. Narayan R, Singh VP, Chakraverty S (2014) Quantum neural network based machine translator for Hindi to English. Sci World J 2014:1–8. https://doi.org/10.1155/2014/485737
60. Sinha RMK, Jain A (2003) Anglahindi: an English to Hindi machine-aided translation system. In: MT Summit IX, New Orleans, USA, pp 494–497
61. Sinha RMK (2005) Integrating CAT and MT in Anglabharti-II architecture. In: 10th EAMT conference, pp 235–244
62. Saha GK (2005) The eb-anubad translator: a hybrid scheme. J Zhejiang Univ Sci A 6(10):1047–1050
63. NCST (2008) Matra: an English to Hindi machine translation system. Technical report, NCST Mumbai
64. Shahnawaz A, Mishra RB (2011) Translation rules and ann based model for English to Urdu machine translation. INFOCOMP J Comput Sci 10(3):25–35
65. Shahnawaz, Mishra RB (2015) An English to Urdu translation model based on CBR ANN and translation rules. Int J Adv Intell Paradig 7(1:1–23
66. Jaideepsinh K, Jatinderkumar S (2016) Sanskrit machine translation systems: a comparative analysis. Int J Comput Appl 136:1–4
67. Huet G (2006) Shallow syntax analysis in Sanskrit guided by semantic nets constraints. In: Proceedings of the 2006 international workshop on research issues in digital libraries. ACM, p 6
68. Kulkarni A, Pokar S, Shukl D (2010) Designing a constraint based parser for Sanskrit. In Sanskrit computational linguistics. Springer, pp 70–90
69. Kulkarni A (2013) A deterministic dependency parser with dynamic programming for Sanskrit. In: Proceedings of the second international conference on dependency linguistics (DepLing 2013), pp 157–166
70. Bhadra M, Singh SK, Kumar S, Agrawal M, Chandrasekhar R, Mishra SK, Jha GN et al (2009) Sanskrit analysis system (SAS). In: Sanskrit computational linguistics. Springer, pp 116–133
71. Kumar A, Mittal V, Kulkarni A (2010) Sanskrit compound processor. In: Sanskrit computational linguistics. Springer, pp 57–69
72. Bharati A, Kulkarni A (2009) Anusaaraka: an accessor cum machine translator. Department of Sanskrit Studies, University of Hyderabad, Hyderabad, pp 1–75
73. Aparna S (2005) Sanskrit to English translator. In: Language in India, vol 5
74. Upadhyay P, Jaiswal UC, Ashish K (2014) Transish: translator from Sanskrit to English-a rule based machine translation. Int J Curr Eng Technol 4(5):2277–4106
75. Gopal M, Mishra D, Singh DP (2010) Evaluating tagsets for Sanskrit. In: International sanskrit computational linguistics symposium. Springer, pp 150–161
76. Gopal M, Jha GN (2011) Tagging Sanskrit corpus using bis pos tagset. In: International conference on information systems for Indian languages. Springer, pp 191–194
77. Gopal M, Jha GN (2007) Indian language part of speech tagger (IL-post). http://sanskrit.jnu.ac.in/corpora/tagset.jsp. Accessed 24 Dec 2018
78. Chandershekhar R, Jha GN (2007) Part-of-speech tagging for Sanskrit. PhD thesis, Special Centre for Sanskrit Studies, JNU Delhi. http://sanskrit.jnu.ac.in/corpora/JNU-Sanskrit-Tagset.htm
79. Sitender Bawa S (2018) Sansunl: a Sanskrit to UNL enconverter system. IETE J Res. https://doi.org/10.1080/03772063.2018.1528187
80. Younger DH (1967) Recognition and parsing of context-free languages in time n3. Inf Control 10(2):189–208
81. Li T, Alagappan D (2006) A comparison of CYK and earley parsing algorithms. In: ICAR-CNR, pp 1–5
82. Papineni K, Roukos S, Ward T, Zhu W-J (2002) Bleu: a method for automatic evaluation of machine translation. In: Proceedings of the 40th annual meeting on association for computational linguistics. Association for Computational Linguistics, pp 311–318
83. LDC (2005) Linguistic data annotation specification: assessment of adequacy and fluency in translations. revision 1.5. Technical report, Linguistic Data Consortium
84. Kumar P, Sharma RK (2012) UNL based machine translation system for Punjabi language. PhD thesis, Thapar University