

An Efficient Long Short-Term Memory Model for Digital Cross-Language Summarization

Y. C. A. Padmanabha Reddy¹, Shyam Sunder Reddy Kasireddy², Nageswara Rao Sirisala³,
Ramu Kuchipudi⁴ and Purnachand Kollapudi^{5,*}

¹Department of CSE, B V Raju Institute of Technology, Narsapur, Medak, T.S, 502 313, India

²Department of IT, Vasavi College of Engineering, Hyderabad, T.S, 500089, India

³Department of CSE, K.S.R.M College of Engineering, Kadapa, A.P, 516003, India

⁴Department of IT, C.B.I.T, Gandipet, Hyderabad, Telangana, 500075, India

⁵Department of CSE, B V Raju Institute of Technology, Narsapur, Medak, T.S, 502 313, India

*Corresponding Author: Purnachand Kollapudi. Email: purnachandkollapudi2022@gmail.com

Received: 05 July 2022; Accepted: 15 September 2022

Abstract: The rise of social networking enables the development of multi-lingual Internet-accessible digital documents in several languages. The digital document needs to be evaluated physically through the Cross-Language Text Summarization (CLTS) involved in the disparate and generation of the source documents. Cross-language document processing is involved in the generation of documents from disparate language sources toward targeted documents. The digital documents need to be processed with the contextual semantic data with the decoding scheme. This paper presented a multilingual cross-language processing of the documents with the abstractive and summarising of the documents. The proposed model is represented as the Hidden Markov Model LSTM Reinforcement Learning ($HMM_{lstm}RL$). First, the developed model uses the Hidden Markov model for the computation of keywords in the cross-language words for the clustering. In the second stage, bi-directional long-short-term memory networks are used for key word extraction in the cross-language process. Finally, the proposed $HMM_{lstm}RL$ uses the voting concept in reinforcement learning for the identification and extraction of the keywords. The performance of the proposed $HMM_{lstm}RL$ is 2% better than that of the conventional bi-direction LSTM model.

Keywords: Text summarization; reinforcement learning; hidden markov model; cross-language; multilingual

1 Introduction

Natural Language Processing (NLP) is an effective platform on computers for the efficient functioning of certain tasks in human languages. It involves the processing of the input provided by the human language in the conversion of the input information into an appropriate representation of the information in another language [1]. The NLP input or output can be either speech or text in the form



This work is licensed under a Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

of information disclosure, language knowledge, semantic, lexical, syntactic, or another knowledge [2]. The NLP process comprises two tasks, such as understanding the natural languages and generating natural languages. At present, the World Wide Web (WWW) is considered a rich and effective source for information processing with a huge development rate of 29.7 billion pages with exponential growth [3]. As per the survey conducted by Netcraft, English is identified as a dominant language on the web [4]. However, the increase in the number of users reveals that non-English users also increased with the information repository, and it is challenging. Those problems with different language users can be overcome with Cross-Lingual Information Retrieval (CLIR) [5].

India comprises of diverse languages around 2000 dialects are identified with stipulated use of Hindi and English for official communication with the national government. India has two official languages used by the national government, as well as 22 scheduled languages for administrative purposes [6]. The identified languages are Bengali, Bodo, Dogri, Gujarati, Hindi, Kannada, Kashmiri, Konkani, Maithili, Malayalam, Manipuri, Marathi, Nepali, Oriya, Punjabi, Sanskrit, Santali, Sindhi, Tamil, Telugu, and Urdu. In addition, English is widely used in India in a variety of contexts, including the media, science and technology, commerce, and education [7]. However, in India, Hindi is considered one of the scheduled languages [8]. Countries like India demand a multi-lingual society involved in the translation of documents from one language to another. Most of the state government documents are based on the regional languages of the Union government and the documents are in bilingual form in either Hindi or English. In order to achieve appropriate communication, it is necessary to convert those documents into languages based on the regional languages [9]. Additionally, regional languages demand the conversion of the daily news as per English through international news agencies. With the human translators, the limitations of information missing in reports and documents are observed. With appropriate machine-based assistance translation or workstation translators, summation is effective for cross-lingual schemes [10].

CLIR is involved in the processing of the languages based on the query languages with other languages for the searched documents. This involved estimation of the information availability in the native languages within CLIR and relevant information about documents being identified [11]. With CLIR, an appropriate relationship is evolved with the information requirements and content availability. The systems are effective for users who evaluate estimates in different languages to gather appropriate information from different language documents to eliminate multiple requests. Cross-language information retrieval [12] facilitates users to search documents in their familiar languages and uses the translation method to retrieve information from other languages, stated as Multilingual Cross Language Information Retrieval (MCLIR). The cross-lingual information access uses the machine learning based translation paradigm of summarization, subsequent translation, and snippets for information extraction from the targeted languages [13]. The CLIR technique uses different techniques, those being a) document translation, b) query translation, and c) inter-lingual translations. Within the CLIR system, to improve performance efficiency, it is necessary to implement knowledge-based robust algorithms for search across many languages and translate the documents.

To improve the functionality of the CLIR, data mining is an effective tool for information processing [14]. Data mining entails estimating non-trivial, implicit information patterns from repositories such as Extensible Markup Language (XML) repositories, data warehouses, relational databases, and so on. At present, large datasets are available for the collection and processing of information better than humans. The evaluation is based on the extraction of the knowledge into a more accurate and information value within the software environment. With larger datasets, traditional techniques are improved for retrieval of information from raw data that is infeasible [15]. The implementation of the data mining concept with machine learning improves the process of CLIR with data selection,

data cleaning, transformation of data, knowledge presentation, and evaluation of patterns. With the core concept of artificial intelligence, machine learning, and statistics, data mining concepts yield benefits of automated information collection and processing. The machine learning technique comprises various methods and techniques with the implementation of different algorithms. In the cross-language information retrieval platform, different techniques are K-nearest neighbour, Nave Bayes, Decision Tree, Neural Network, and Cluster analysis [16].

Contribution of the Work

In search of the World Wide Web, English is considered a primary language, with an increase in the number of users. Non-English native speakers also search for documents. To facilitate the ease of search for people, it is necessary to construct an appropriate domain with machine learning for identification and translation of the cross-language abstraction and summation model. The proposed model $HMM_{lstm}RL$ comprises the LSTM model with HMM integrated with reinforcement learning. The specific contribution is presented as follows:

- With the Hidden Markov model, word count and number of keywords count are considered for analysis and processing. The LSTM model calculates the total number of words in the statement or document. Through a calculated number of words, the number of repeated words is updated in the neural network model.
- To develop a bi-directional LSTM-based corpus model for multi-language encoding processing with keyword extraction.
- To construct a reinforcement learning based machine learning model for feature extraction and word identification. Upon the testing and validation of the information, words are processed and updated on the network.
- Within reinforcement learning, the MapReduce framework is applied for the clustering and removal of the same words in the text document. Finally, voting is integrated for the abstraction and summarization of the keywords.
- The experimental analysis showed that the proposed $HMM_{lstm}RL$ achieves higher precision and recall value compared with the conventional techniques.

This paper is organised as follows: Section 2 investigates how the cross-lingual processing model works. In Section 3, research methods are adopted for the encoding and decoding of data in multi-lingual systems with the proposed $HMM_{lstm}RL$. In Section 4, experimental analysis of the proposed $HMM_{lstm}RL$ model is comparatively examined with existing techniques. Finally, in Section 5, the overall conclusion is presented.

2 Related Works

The key challenge for English to Hindi statistical machine translation is that the Hindi language is richer in morphology than the English language. There are two strategies that facilitate reasonable performance in this language pair. Firstly, reordering of English source sentences in accordance with the Hindi language is the second strategy, which is by making use of suffixes of Hindi words. Either of these strategies or both strategies can be used during translation. The difference in word order between the Indian language and English makes these two strategies equally challenging. For example, the English sentence “he went to the office” has the Hindi translation “vah (he) kaaryaalay (office) gaya (went to).” In this example, it is evident that the position of words in English is not retained in the translated Hindi text. An author has developed an unsupervised part-of-speech tagger that makes use of target language information, and it has been proven that the results are better as compared with

the Baum-Welch algorithm [17]. The main demerit of this approach is the increase in the required number of translations in an exponential manner. Thus, increasing the overall time of execution due to the increase in time for translation. The pruning method was used by the author to identify the unlikely disambiguation. Thus, reducing the number of translations required and also the overall time complexity of the algorithm.

In [18] proposed interpretable Hidden Markov Models (HMM)-based approaches for emotion recognition in text and analysed their performance under different architectures (training methods), ordering, and ensembles (e.g.). The presented models are interpretable; they may show the emotional portions of a phrase and explain the progression of the overall feeling from the beginning to the end of the sentence. Experiments show that the new HMM algorithms and training methods are just as good as machine learning methods and, in some cases, even do a better job than older HMMs.

In [19] developed a cross-language text summarization (CLTS) technique to generate word summaries of different languages from different documents. The compressive CLTS approach examined the source text and target language for the computation of relevant sentences. The developed system comprises two-level sentences such as clustering of similar sentences with multi-sentence compression (MSC). The second technique is the use of compressed neural network models. The developed approach comprises the compression of the multi-sentence generated with the French-to-English summarization information for the extractive state of the art system. Moreover, the developed cross-lingual summarization exhibits the desired grammatical quality with the extractive approaches.

In [20] proposed a framework for text summarization based on group with semantic clustering with text summarization based on grouping related sentences with the Semantic Link Network (SLN) based on group ranking with the summary. With the implementation of the SLN model with the two-layer model with semantic links, with part of a link, a sequential link, similar to a link, and a link for cause and effect, the experimental analysis expressed that the composed summaries involved in groups or paragraphs with composed sentences and summaries had an average source text length of 7000 words to 17,000 words usual length. Furthermore, seven clustering algorithms are generated and grouped with the five strategies through semantic links.

In [21] aimed to increase the Arabic summarisation accuracy through the novel topic-aware abstractive in the Arabic (TAAM) summarisation model with the integrated recurrent neural network. The quantitative and qualitative analysis of the TAAM model uses the ROUGE matrices, which provides a higher accuracy value of 10.8% than the baseline models. Additionally, with the TAAM model, it generates coherent Arabic summaries that are captured and read for the input idea.

In [22] constructed a problem-based learning approach for natural language processing (NLP) processing in undergraduate and graduate students. The team of students uses sets for big data, guidelines, resources for cloud computing and other aids for the different teams with the collection of two big data those are related to web pages and electronic theses and dissertations (ETDs). The team of students is deployed based on consideration of the different methods, tools, libraries, and tasks for summarization. As the summarization is a process involved in addressing the NLP learning process for the different linguistic levels based on consideration of NLP practitioners' tools and techniques. The results evaluation revealed that coherent teams are generated based on readable summarization. With different summarization techniques, the ETD process is evaluated through high quality and accurate utilisation of the NLP pipelines. Additionally, the developed technique expresses that the data and information are managed effectively in those fields for the developed approach or similar methods for the defined data sets through the provision of synergistic solutions.

In [23] presented a narrative abstractive summarization (NATSUM) technique for the chronological ordering of the target entity based on the consideration of the chronological order within the same topic. To achieve an effective cross-summarization process, a timeline for the cross-document is mentioned for the different point events for the same event. With the timeline-based arguments, the documents' events are extracted and processed. Secondly, with the natural language generation, sentences are produced with event arguments. Specifically, through the realisation technique, hybrid surfaces are derived based on the technique for the generation and ranking of hybrid surfaces. The experimental analysis showed that the proposed NATSUM exhibits effective summarization through abstractive baselines to increase the F1-measure by 50% for a simulated environment.

In [24] proposed a framework for the syntactic features using novel Syntax-Enriched Abstractive Summarization (SEASum) with graph attention networks (GATs) with the implementation of the source texts. With the implemented SEASum architecture, the sequence of the Pre-trained Language Models (PTM) semantic encoder is implemented with the graph attention networks (GAT)-based explicit syntax for the source document presentation. With the summarization framework, a feature fusion module is introduced through the syntactic features for the summarization models using the SEASum model for the semantic and syntactic encoder for multi-head fusion of the feature stream in the decoding process. Secondly, through the SEASum cascaded model, contextual embedding is performed for the semantic words for information flow. The experimental results show that the built SEASum and cascaded models outperform the abstractive summarization approach.

3 Hidden Markov Model for the Extraction of Features

The proposed HMM_{istm}RL comprises of the coding scheme applied with the Hidden Markov Model for abstraction and summation. Initially, the character is estimated with HMM_{istm}RL perform coding scheme for the estimation of the stored characters in the computer in the form of bits. The detection of the charset is based on estimation of the Unicode transformation (UTF – 8) due to presence of larger bytes in the sequences. The bytes are computed based on the implementation of the bytes with validity test through *extremely* unlikely scenario. With heuristics detection scheme it is necessary to assign label properly in the dataset label with appropriate encoding mechanism. Through hypertext markup language (HTML) documents encoding is in the meta elements as <Metahttp-equiv="Content-Type" content="text/html; charset=UTF-8"/>. In sequence, the document is processed with the Hypertext Transfer Protocol (HTTP) with the similar meta data for the content type header. Finally, the encoding is applied over the text filed for the initial bytes label explicit. The Unicode model comprises of the different unique number for each character those are stated as follows:

Accommodates more than 65,000.

Synchronized with the corresponding versions of ISO-10646.

Standards incorporated under Unicode

ISO 6937, ISO 8859 series

ISCII, KS C 5601, JIS X 0209, JIS X 0212, GB 2312, and CNS 11643 etc.

Scripts and Characters, European alphabetic scripts

Middle Eastern right-to-left scripts, Scripts of Asia

Indian languages Devanagari, Bengali, Gurmukhi, Oriya, Tamil, Telugu, Kannada, Malayalam.

Punctuation marks, diacritics, mathematical symbols, technical symbols, arrows, dingbats, etc.

The text processing system comprises of the dependent components based on the dictionaries in the targeted language document with the assigned character codes. The each elements code is defined as Unique number code point with the hexadecimal prefix number of U“Ex”, U+0041 with value of “A”. The language character level is identified and processed with the statistical language identification for the training data in reinforcement learning [25–31]. The performance is evaluated based on the consideration of the characteristic’s samples defined as in Table 1.

Table 1: Characteristics of languages

Language	Unigram Character Set
Kannada	e, , a, , so on
English	A, B, C, D, so on
Telugu	అ ఆ ఇ ఐ so on

3.1 Hidden Markov Model for Position Estimation

Hidden Markov model has fewer assumptions of independence. In particular, HMM does not assume that probability that condemnation i is in summary is independent of whether condemnation $i-1$ is in summary. The syntactic and semantic features are extracted from the source language text and are used in the transfer phase to generate the sentence in the target language. This information is extracted by the HMM source language analysis phase. This phase is further sub-divided into–part-of-speech tagging and word sense disambiguation. Once this information is extracted, it is used in the transfer phase which makes use of Bayesian approach. This sense-based machine translation system makes use of Bayesian approach, which is based on the statistical analysis of existing bilingual parallel corpora. The label assigned with the HMM model is defined as follows based on features.

Feature Name	Label	Value
Position of Paragraph	O1	1,2,3
Number of terms	O2	$\log(w_i+1)$
Baseline Term Probability	O3	$\log(\text{Pr}(\text{terms in } i \text{baseline}))$
Document Term Probability	O4	$\log(\text{Pr}(\text{terms in } i \text{document}))$

The position of condemnation in its paragraph. We assign each condemnation value $o1(i)$ designating it as rest in paragraph (value 1), last in paragraph (value 3), or intermediate condemnation (value 2) condemnation in one-long paragraph is assigned value 1, & condemnation s in two-long paragraph are assigned values of 1 & 3. A number of terms in condemnation. value of this feature is computed as $o2(i) = \log(\text{number of terms} + 1)$.

The content processing with the HMM_{istm}RL with application of the HMM is defined as

1. The position of condemnation is incorporated with state-structure of HMM.
2. Estimation of this component is $o1(i) = \log(\text{number of terms} + 1)$
3. Likely condemnation terms are, given report terms $o2(i) = \log(\text{Pr}(\text{terms in condemnation } i|D))$.

The statistical analysis is also performed based on the local syntactic information available in the input sentence. Using the analysed data, the Bayesian approach is applied to predict the probable target word for the given input word. This proposed word sense-based statistical machine translation system may be mathematically expressed as in Eq. (1)

$$P\left(\frac{w_t}{w_t Tags_s}\right) = \left(\frac{w_s Tags_s}{w_t}\right) * P(w_t) \quad (1)$$

In the above Eq. (1), source word is denoted as w_s ; part of the source word is defined as $Tags_s$ and target word is denoted as w_t . The probable target word w_t has dependency on the source word w_s and its part-of-speech $tags$. Thus, to find the probable target word Wt with the prior knowledge about the input source word and its predicted part-of-speech, a Bayesian approach is used. Fig. 1 shows the pre-processing phase in proposed method.

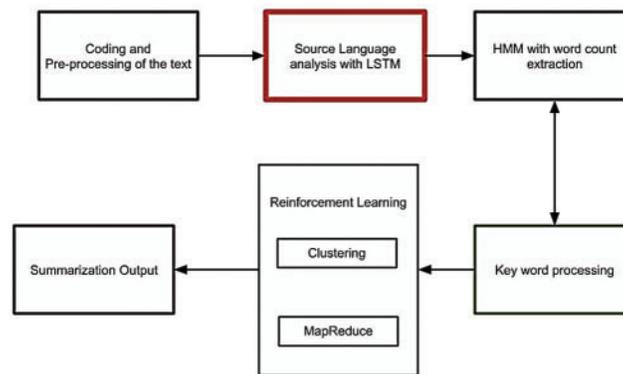


Figure 1: Pre-Processing phase in $HMM_{istm} RL$

Since the proposed machine translation system works at word level, there is need for tokenization of source text at sentence level as well as word level. In Hindi language, there is tense, aspect and modality (TAM) information stored in the affixes of the words. These affixes also contribute to the accuracy of machine translation. To extract the TAM information stored in the affixes, longest affix matching algorithm is used to check the matching between affixes. Levenshtein distance is used to calculate the matching score. For example, consider the word खाने (khaane) and रहेगा (rahega) mentioned in input sentence – “रात को एक रोटी कम खानेसेपेट हल्का रहेगा । (raat ko ek rotee kam khaane se pet halka rahega)”. Both these words have TAM information in it and it can be extracted using the longest affix matching algorithm. The word खाने (khaane) will be analyzed and the TAM information is found as masculine, plural verb. Similarly, the TAM information of the word रहेगा (rahega) is found to be masculine, singular verb. Once these affixes are extracted, the sequence of words along with its affixes are fed to the next phase of the translation system as shown in Table 2.

Table 2: Estimation of affix in statement

Hindi Word	Part-of-speech	English Equivalent	Tamil Word
खाना (khana)	Noun	Food	சாப்பாடு (cappadu)
खाना (khana)	Verb	Eat	சாப்பிட (Cāppiṭa)
नीला (neela)	Adjective	Blue	நீலம் (Nīlam)
कर (Kar)	Verb	Do	சசய் (Cey)
आम (Aam)	Noun	Mango	மாம்பழம் (Māmpaḷam)
साफ (Saf)	Adverb	Clean	சுத்தமான (Cuttamāṇa)

3.2 Word Count Extraction with Bi-Directional LSTM in $HMM_{lstm}RL$

In the LSTM model, the probability value of the document drivers is calculated from the existing languages. The probability is computed based on the occurrence of the string value as S with the alphabet X sequence is represented in Eq. (2). The defined probability for the string S in the character sequence for the words in document is represented in Eq. (3).

$$p(s) = p(s_1, \dots, s_n) \quad (2)$$

$$P_{im} = \frac{c_{im}}{\sum_{i=1}^N C_{im}} = \frac{c_{im}}{c_m} \quad (3)$$

c_m – words total number occurred in the word m from the collection of N documents

P_{im} – in the document i number of words m appear

With the predefined categories for n values differentiate keywords are assigned with category c_i for every other category value of c_j . In the LSTM model the combination of the probabilities in the language identified is estimated based on each letter. The selected language in the document highest probability is computed using the Eq. (4)

$$nc_r P^r (1 - P)^{n-r} \quad (4)$$

The distribution of parameters is based on the consideration of the different variable n with the trails number and probability of occurrence of character in the document Unigram as p. The multilingual corpus texts for the characters are understand with the character set defined as follows:

ಅ, ಆ, ಇ, ಈ, so, on are charset of Kannada, A, B, C, D, so on are charset of Latin English
అ ఆ ఇ ఈ so on are charset of Telugu.

3.3 Text Extraction and Clustering with MapReduce

The hidden markov model-based MapReduce tagger is a statistical approach which is used to identify the probable part-of-speech of each word in the sentence. The hidden markov model-based tagger basically finds the most probable sequence of part-of-speech for a given sentence by using the transition probability $P(W_n/T_n)$ and emission probability $P(T_n/T_{n-1})$. These transition and emission probabilities are learned by the tagger using the monolingual Hindi corpus. These probabilities are calculated using the expressions as in Eqs. (5) and (6),

$$P\left(\frac{W_n}{T_n}\right) = \frac{\text{Count}(w_n, T_{nc})}{\text{Count}(T_n)} \quad (5)$$

$$P\left(\frac{T_n}{T_{n-1}}\right) = \frac{\text{Count}(T_n, T_{n-1})}{\text{Count}(T_{n-1})} \quad (6)$$

The Input word sequences are denoted as W1, W2, W3, ..., n and the part-of-speech of each word is denoted as T1, T2, T3, ..., Tn. The part-of-speech of each word T1, T2, T3, ..., Tn acts as hidden states. Each of these hidden states is predicted using the emission and transition probabilities. For example, consider the input sentence as “रात को एक रोटी कम खानेसेपेट हल्का रहेगा। (raat ko ek rotee kam khaane se pet halka rahega.)” and the POS tagset as [JJ (adjective), N_NN (Noun), PSP (Postposition), QT_QTC (Cardinal), QT_QTF (Quantifiers), V_VM (Verb), RD_PUNC (Punctuations), CC_CCS (Conjuncts)]. The beginning of a sentence is denoted by ^ symbol. Considering the first word “रात (raat)” in the sentence to calculate the probability of the word as a noun,

$$P\left(\frac{\text{रात}}{N_NN}\right) = \frac{\text{Count}(\text{रात}, N_NN)}{\text{Count}(N_NN)} \quad (7)$$

$$P\left(\frac{N_NN}{\wedge}\right) = \frac{\text{Count}(N_NN, \wedge)}{\text{Count}(\wedge)} \quad (8)$$

The frequency of occurrence count is found from the monolingual Hindi corpus. Consider there are 35 occurrence of word रात (raat) as noun (N_NN) and the occurrence of noun (N_NN) to be 1000 out of which 600 times it is tagged with first word of the sentence. Number of sentences is also considered as 1000.

$$P\left(\frac{\text{रात}}{N_NN}\right) = \frac{\text{Count}(\text{रात}, N_NN)}{\text{Count}(N_NN)} = \frac{35}{1000} = 0.035 \quad (9)$$

$$P\left(\frac{N_NN}{\wedge}\right) = \frac{600}{1000} = 0.6 \quad (10)$$

In similar manner as in Eqs. (9) and (10) each word probability is computed in the LSTM network. The computed words in the documents are aligned and processed as stated in Table 3.

Table 3: Word alignment in HMM_{lstm} RL

Hindi Word	Hindi Word Position	Tamil Word	Tamil Word Position
पानी (paanee)	1	நீர் (nīr)	2
पीना (peena)	2	குடி (kuḍi)	1
अच्छा है। (achchha hai.)	4	நல்லது. (nallatu)	4

Hindi Text: पानी पीना स्वास्थ्य केवलए अच्छा है। (paanee peena svaasthy ke lie achchha hai.)

English Equivalent: Drinking water is good for health.

Tamil Text-1: குடிநீர் ஆவராக்கியத்திற்கு நல்லது.

Tamil Text-2: ஆவராக்கியத்திற்கு நல்லது குடிநீர்.

Tamil Text-3: நல்லது ஆவராக்கியத்திற்கு குடிநீர்.

To disambiguate the appropriate sense of the word mentioned in source text, a word sense disambiguation is used.

The identified senses are used in the proposed statistical machine translation approach. The transfer phase basically predicts the probable target language word based on the source language word and its part-of-speech. Using Bayes rule, the transfer phase is mathematically expressed as below in Eqs. (11)–(14)

$$P\left(\frac{w_t}{w_s, Pos}\right) = P\left(\frac{w_s pos}{w_t}\right) * P(w_t) \quad (11)$$

$$P\left(\frac{w_s pos}{w_t}\right) = \left(\frac{w_s}{w_t}\right) * P\left(\frac{pos}{w_t}\right) \quad (12)$$

$$P\left(\frac{W_s}{w_t}\right) = \frac{Count(w_n, wt)}{Count(w_t)} \quad (13)$$

$$P\left(\frac{pos}{w_t}\right) = \frac{Count(pos, wt)}{Count(w_t)} \quad (14)$$

The target word (W_t) is conditionally dependent on its previous word (W_{pre}) in the text. Similarly, every target word has dependency with its preceding word. So, the probability of target word [60] is defined as a n-gram model and is mathematically expressed as in Eqs. (15) and (16)

$$P(w_t) = \frac{P(w_t)}{W_{prev}} \quad (15)$$

$$P(w_t) = \frac{count(wt, W_{prev})}{count(W_{prev})} \quad (16)$$

3.4 Reinforcement Learning with MapReduce for Word Summation

In this proposed HMM_{istm}RL model, the part-of speech (pos) of the source text is also considered as a parameter to predict the probable alignment for a word. Thus, the probable position of target word can be calculated using its conditional dependence with position of source word, length of source text, length of target text and part-of speech of source word. Using Bayes theorem, the modified word alignment model is mathematically represented in below Eq. (17)

$$P\left(\frac{j}{i, l, m, pos}\right) = \frac{P(i, j, l, m, pos)}{P(i, l, m, pos)} \quad (17)$$

In the above Eq. (17), target word position is represented as j, source word is denoted as i, source text of the word is represented as l, part of speech is denoted as pos and target text as m. Fig. 2 shows the overall flow of proposed method.

The hidden layer in the MapReduce tagger is activated using a sigmoid function which is expressed mathematically as below,

$$y(f(x)) = \frac{1}{1 + e^{-f(x)}} \quad (18)$$

$$f(x) = \sum_{j=1}^N w_j * P\left(\frac{word}{tag_j}\right) * P\left(\frac{tag_j}{tag_i}\right) \tag{19}$$

word – current word being processed
tag_i – Part-of-speech tag of previous word
tag_j – Part-of-speech tag from the considered POS tag set
N – number of distinct part-of-speech tags

The abstraction and summation phase of the proposed HMM_{lstm}RL is estimated with reinforcement learning defined as in Eq. (20)

$$P\left(\frac{w_t}{w_s, Pos}\right) = P\left(\frac{w_t}{w_s, pos}\right) * P\left(\frac{w_t}{w_s, pos}\right) \tag{20}$$

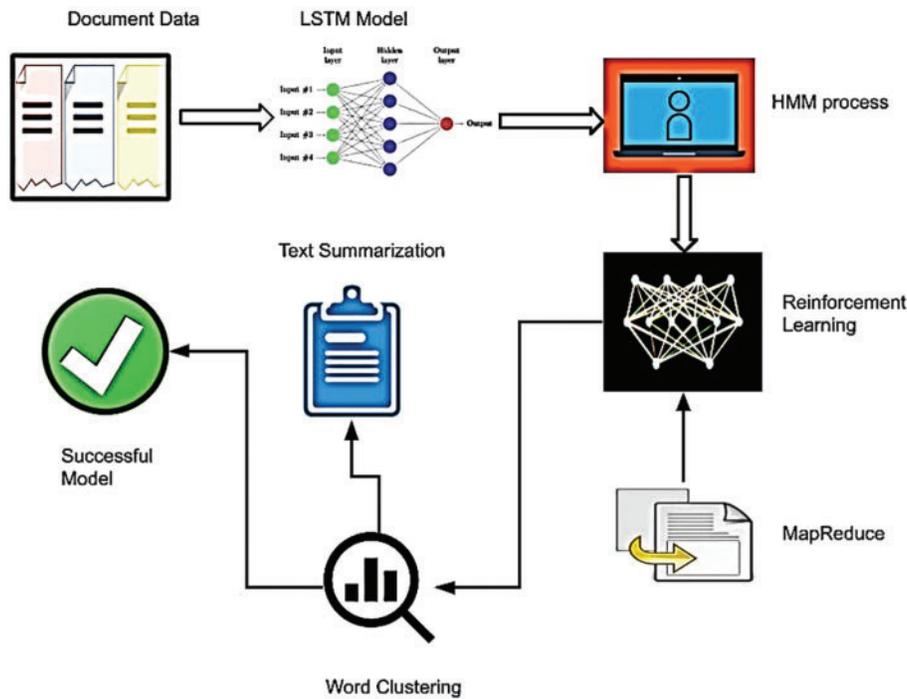


Figure 2: Overall flow of the HMM_{lstm}RL

Using extended Bayes theorem, the above expression (20) is rewritten as,

$$P\left(\frac{w_t}{w_s, Pos}\right) = \left[P\left(\frac{w_s}{w_s}\right) * P\left(\frac{pos}{w_p}\right) P\left(\frac{w_p}{w_p - 1}\right) \right] * \left[P\left(\frac{w_p}{w_t}\right) * P\left(\frac{pos}{w_t}\right) P\left(\frac{w_t}{w_t - 1}\right) \right] \tag{21}$$

The HMM_{lstm}RL approach for Hindi to Tamil machine translation is compared with the naïve Bayes statistical machine translation system in terms of the features that are being used in both the system. The term-document frequency matrix is constructed using all the above sentences [S1, S2, . . . , S6] along with the input sentence. Using these 7 sentences, the number of distinct words is identified as 39. Thus, the term-document frequency matrix (A) will be of size (39x7) and is as shown below,

$$A = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (22)$$

The cosine similarity between vectors in right singular matrix and first row of singular diagonal matrix is calculated. The resultant vector after applying cosine similarity is as below,

$$\text{Cosine similarity} = [0.13 \ 0.3341 \ 0.5423 \ 0.4866 \ 0.5335 \ 0.4971 \ 0.5086]$$

The sentence which has the cosine similarity nearer to 0 is closest to the input sentence. The first value in the vector denotes the cosine similarity with the input sentence itself and it is natural that it will be closest to zero. The next smallest value in the vector is 0.3341 which is the cosine similarity value of sentence S1.

3.5 Word Extraction in the Sentence

This paper concentrated on the evaluation of language keywords through the implementation of a stacked classifier integrated with a voting scheme. This research utilizes machine learning for language estimation and classification. This research considers four classifiers such as AdaBoost, Artificial neural networks (ANN), Decision tree, and Support Vector Machine (SVM) integrated with the voting scheme. The developed HMM_{lstm}RL is adopted through machine learning, which involves several steps those are input data, data pre-processing, feature extraction, feature selection, and classification. Based on the proposed HMM_{lstm}RL classification of language classification. The steps implemented in the proposed mechanism are presented as follows:

Input Data: The dataset collected from different data sources those are incorporated in machine learning for data processing.

Data Pre-processing: In this stage collected dataset is evaluated for the elimination of redundant data.

Feature Extraction: In this stage, language features are extracted for further processing of machine learning.

Feature Selection: It eliminates the irrelevant features from the set of extracted data. To eliminate the irrelevant features for minimization of computational complexity with improved accuracy.

Classification: This stage provides the results of the present data. This is adopted two factors such as training and testing:

Training: This stage of machine learning involved training machine learning algorithms for making computers learn from the extracted features for the languages. Based on the classifier parameters are trained to machine learning to fit the classifier.

Testing: The dataset for classification is evaluated based on the evaluation of specified classifier.

In the proposed HMM_{lstm}RL mechanism collected data features are processed for machine learning stages. The proposed HMM_{lstm}RL scheme uses 4 classifiers with stacking in machine learning. The 4 classifiers considered are AdaBoost, ANN, decision tree, and SVM in the machine learning process in HMM_{lstm}RL. The proposed HMM_{lstm}RL is involved in the classification of attacks. Initially,

language classification is performed with consideration of 4 classifiers such as AdaBoost, SVM, decision, and ANN. With classification, if it is identified as an attack for 2 classifiers and not attack as another classifier then the proposed $HMM_{Istm}RL$ evaluates with a decision tree. The decision tree algorithm is involved in the decision-making of the language whether the file is an attack or not. Based on the classification results provided by the decision tree network system will estimate which language it belongs. The classification of the language is based on the voting mechanism.

Initially, the proposed $HMM_{Istm}RL$ train the model through classifiers such as AdaBoost, ANN, SVM, and decision tree. The model trained through the classifier is exported to the predictor for the computation of language. The predictors evaluate the predictor vector of each classifier and update to voting. The classification voting approach is based on the score value of the classifier; if the value obtained from the four classifiers is computed as a language, then the network is considered another language keywords. In case if two classifiers are stated as a keyword for the language then the proposed $HMM_{Istm}RL$ goes with the decision tree process for computation. To make a decision proposed $HMM_{Istm}RL$ utilizes a voting mechanism. The voting scheme is utilized for the estimation of languages. The analysis is based on the summation of classifier value for computation of languages belongs to keyword or not. The voting scheme is estimated based on the score of the classifier through consideration of condition such as:

if score > 2; then keyword language
 else 0; other language

Based on the computation of classifier value voting score is computed. Through computed voting score proposed $HMM_{Istm}RL$ compute the data is belongs to particular language or not. Fig. 3 shows the voting process in $HMM_{Istm}RL$.

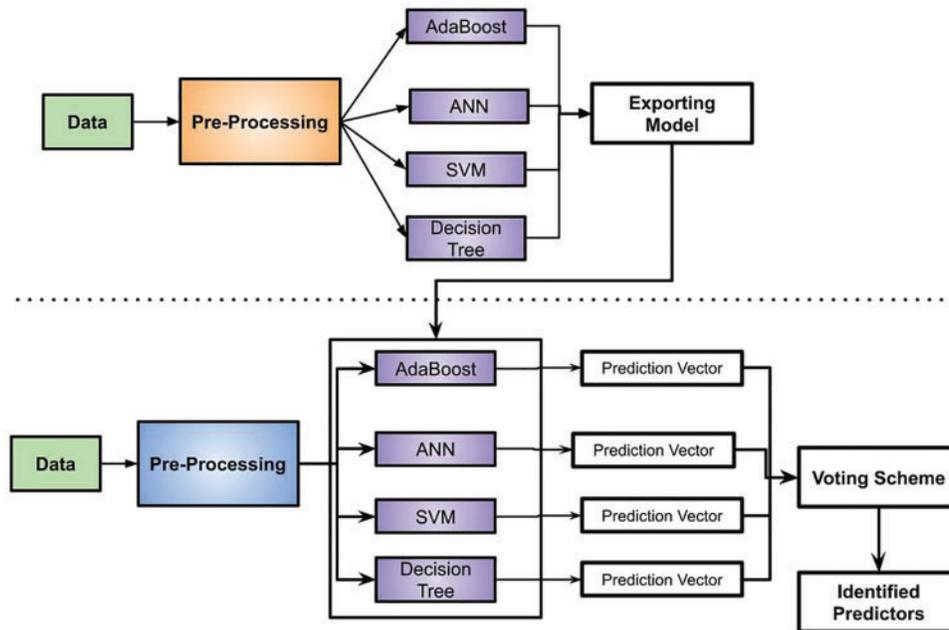


Figure 3: Voting process in $HMM_{Istm}RL$

4 Results and Discussions

The proposed HMM_{lstm}RL comprises the sequence-to-sequence model and was developed using pytorch with TensorFlow at the backend. TensorFlow is an open-source platform used for developing deep neural networks. The proposed model learns the features from the input vector and target vector. These features are used to generate the target text based on the input vector fed to it. To make the model learn the features in an efficient way, there is a need for a huge amount of corpus. Due to this feature, a sentence in Tamil and Hindi can be shuffled in different combinations to generate variants of the given sentence. Since the Hindi language is a partially free word order language, all the combinations generated will not be grammatically correct. Thus, there is a need for verifying the grammatical correctness of the text being generated. Parsing the sentences will be helpful in checking the grammatical correctness of them. In this proposed approach, HMM_{lstm}RL is used to verify the correctness of the generated Hindi sentence. The Hindi parser verifies the grammar by parsing the tagged text fed to it. In this way, the valid variants of Hindi text are generated along with its Tamil sentences and are maintained in the training dataset. Similarly, the Tamil sentences can also be shuffled, but there is no necessity for verification of grammar in them. This is due to the fully free word order nature of the language.

The proposed HMM_{lstm}RL model was analysed with various dropout percentages and an optimal percentage value was found to be in the range of 20% to 60%. Since there is an encoder module and a decoder module, there is a need to analyse the dropout percentage in both these modules such that the performance of the overall system is good. The ideal dropout percentage for encoders is found to be 20%, and the ideal dropout percentage for decoders is 60%. The following are parameters that were used for the sequence-to-sequence model,

Number of epochs = 22

Learning rate = 0.01

Hidden layer size = 2

Dropout = 0.2 (in encoder) and 0.6 (in decoder)

After 22 epochs, the feature learning by model gets saturated and the necessity for training becomes negligible. The learning rate is kept at 0.01 and the accuracy of the model fluctuates when the learning rate is made a 0.1. This is due to rapid change on the weights and thus the model was not able to learn the features. The number of hidden layers used in the proposed approach is two. The performance of proposed model is comparatively better as compared with the model having more than two hidden layers.

For the analysis taking an example from our published word [61], consider the input text as: “ताजा साँसेँ और चमचमाते दाँत आपके व्यक्तित्व को वनखारते हैं।” (English Equivalent: Fresh breath and shining teeth enhance your personality. After performing the initial pre-processing phase, the words in the input sentence are identified as unique token and these identified tokens are further used to extract the syntactic information in the sentence. The syntactic information is extracted using HMM based part-of-speech tagger. The sample part-of-speech tags used in this input sentence are described in [Table 4](#).

The tagged output for the text considered will be as below, ताजा\JJ साँसेँ\N_NN और\CC_CCD चमचमाते\JJ दाँत\N_NN आपके\PR_PRP व्यक्तित्व\N_NN को\P SP वनखारते\V_VM हैं\V_VAUX I.\RD_PUNC. (taaja\JJ saansen\N_NN aur\CC_CCD chamachamaate\JJ daant\N_NN aapake\PR_PRP vyaktitv\N_NN ko\PSP nikhaarate\V_VM hain\V_VAUX I.\RD_PUNC.) This tagged

text is fed to the word sense disambiguation phase to identify the various possible senses for the word used in that context. Considering the word ताजा\JJ (taaja\JJ), the various senses for this word is retrieved from the Hindi wordnet. The various senses retrieved are – ताजा (taaza), ताजा (taaja), अमलाान (amalaan), अशशुक (ashashuk), आला (aala), गरमागरम (garamaagaram), टटका (tataka). For all the retrieved senses, its corresponding usage sentences are retrieved which are used in the word sense disambiguation phase to identify the senses that match with the input text. The identified possible senses for the word ताजा are ताजा and ताजा as shown in Table 5.

Table 4: Computation of tags in sentence

Tag	JJ	N_NN	CC_CCD	PR_PRP
Part-Of-Speech	Adjective	Noun	Conjunction	Pronoun
Tag	PSP	V_VM>	V_VAUX	RD_PUNC
Part-Of-Speech	Postposition	Finite Verb	Auxiliary Verb	Punctuation

Table 5: Estimated sentences with HMM_{istm}RL

Target Text (generated):	சூயிங்கம் (cuyinkam)	சமல்லுவதினால் (melluvatinal)	உமிழ் நீர் (umilnir)	உற்பத்தியாகின்றது. (urpattiyakinratu)	<EOS>
Reference Text	சூயிங்கம் (cuyinkam)	சமல்லுவதினால் (melluvatinal)	உமிழ் நீர் (umilnir)	உற்பத்தியாகின்றது. (urpattiyakinratu)	<EOS>
Matching	1	1	1	1	1

The proposed HMM_{istm}RL compute sequence to sequence model was tested with various test set. The generated target sentences were evaluated using Bilingual evaluation understudy (BLEU) score. The Table 6 shows the analysis of BLEU score with different training and testing pairs. The BLEU score calculated for this proposed system is the average of sentence level BLEU score. The BLEU score is found to improve with increase in the percentage of training set being used. The accuracy of proposed model is good when the training and testing corpus ratio is kept at 80:20. The estimation of machine learning parameters is shown in Table 6.

Table 6: Estimation of machine learning parameters

Training/Testing Corpus Size (in %)	BELU Score
60/40	0.7037
70/30	0.7234
80/20	0.7588
90/10	0.6628

The neural machine translation system is also evaluated at different runs by keeping the ratio of training and testing pair as 80:20 as shown in Table 7. At each of the runs the training and testing set has been changed and are chosen at random. It is also found that the neural machine translation system has accuracy better than the statistical machine translation system.

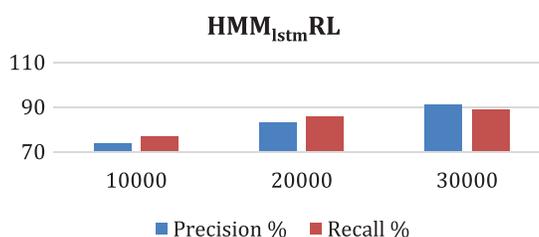
Table 7: Computation of BELU values

Number of Runs	BELU score
1	0.7478
2	0.7176
3	0.6784
4	0.7118
5	0.7588
6	0.7124
7	0.7022
8	0.6914
9	0.7156
10	0.7211

The word embedding is performed using a continuous bag-of-words model and it is found to capture the semantics in the words. This in turn helped in improving the accuracy of the translation using the sequence-to-sequence model. Since Hindi and Tamil language are morphologically rich, there is need for semantic mapping which is made using this approach. The results are found to be far better than any state-of-art method for these two languages. It is found that BLEU score is 0.7588 and it can be improved further by using a properly aligned parallel corpora as shown in Table 8. Fig. 4 shows the proposed HMM_{lstm}RL recall and precision.

Table 8: Performance analysis

Corpus Size (In number of words)	HMM _{lstm} RL		Without HMM _{lstm} RL	
	Precision (in %)	Recall (in %)	Precision (in %)	Recall (in %)
10000	74	77	56	51
20000	83	86	70	73
30000	91	89	79	82

**Figure 4:** HMM_{lstm}RL recall and precision

The target word is thus predicted to be “ஹுடா (podhu)” for the English word “general”. Similarly, all other words are translated to its probable Tamil word. In this case, the grammar of the input

sentence matches with the target text. Thus, there is no need for structural rearrangements. This is identified by using the naïve Bayes approach described before.

The generated sentence is compared with the reference sentence and is as shown in Table 9. The proposed system has been analyzed on various other source sentence and its evaluation metrics are calculated. The results of evaluation metrics are as shown in Table 10. Which details about the precision and recall of the proposed pivot language-based system. The table also shows the comparison of proposed system with the statistical machine translation without pivot language.

Table 9: Comparison of generated target text with its reference text

Generated Text	ஒரு	சபாஈ	நபரின்	ரத்த	அழுத்தம்	140 –	குளறக்க	வவண்டும்.
	(oru)	(podhu)	(naparin)	(iratta)	(aluttam)	90	(kuraikka)	(ventum)
Reference Text	ஒரு	சாதாரண	நபரின்	இரத்த	அழுத்தம்	140 –	குளறக்க	வவண்டும்.
	(oru)	(catarana)	(naparin)	(iratta)	(aluttam)	90	(kuraikka)	(ventum)
Match	1	0	1	1	1	1	1	1

Table 10: Comparison of proposed system with the statistical machine translation without pivot language

S.No.	Corpus Size (In number of words)	Bi-directional LSTM		Proposed HMM _{lstm} RL	
		Precision (in %)	Recall (in %)	Precision (in %)	Recall(in%)
1	10000	53	52	58	57
2	20000	64	67	71	72
3	30000	76	74.5	83	79

The performance of the bi-directional LSTM seems to be poor as compared with the statistical machine translation without pivot language which is evident from the graph shown in Fig. 5. The BLEU score of this pivot-based system is found to be 0.54.

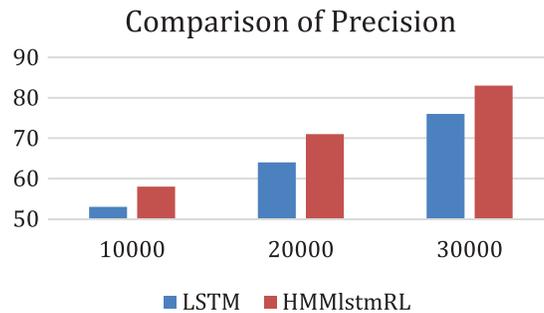


Figure 5: Comparison of Precision

Based on the above calculation, it is found that the word சாதாரண (Cātāraṇa) will be most probable translation for the word सामान्य (saamaany). The structural transfer phase has identified that the sequence does not need any rearrangements to retain the target language grammar. The final

translated text for this input sentence is shown in Table 11. And it is also compared with the reference sentence mentioned in the corpus. Fig. 6 shows the Recall of the proposed method.

Table 11: Analysis of generated target text with its reference text

Generated Text	ஒரு சாதாரண நபரின் இரத்த அழுத்தம் 140 – குளறக்க வவண்டும்.
	(oru) (catarana) (naparin) (iratta) (aluttam) 90 (kuraikka) (ventum)
Reference Text	ஒரு தாரண நபரின் இரத்த அழுத்தம் 140 – ளறக்க வவண்டும்.
	(oru) (catarana) (naparin) (iratta) (aluttam) 90 (kuraikka) (ventum)
Match	1 1 1 1 1 1 1 1

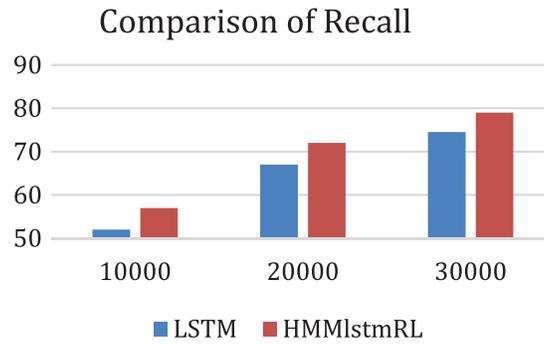


Figure 6: Comparison of Recall

From Table 12, the target sentence is found to be the same as it has to be with respect to the reference sentence. The proposed $HMM_{lstm}RL$ machine translation system has been evaluated using various sentences (restricted to health domain) and the Bilingual Evaluation Understudy (BLEU) score is calculated to be 0.68. The corpus size when increased, further leads to increase in distortion noise and thus producing a poor translation accuracy. It has been found ideally that the unique word should be kept at 30000 and further increase in the corpus size leads to poor translation due to increase in noise. The proposed system is also compared with the other statistical machine translation system in terms of BLEU (Bilingual Evaluation Understudy) score, which is shown in Table 13. The $HMM_{lstm}RL$ approach is found to have a BLEU score of 0.7637 and it is also noted that the BLEU score has improved by few percentages when compared with the pivot-based approach.

Table 12: Comparison with different corpus sizes

S. No.	Corpus Size (In number of words)	Bi-directional $HMM_{lstm}RL$		Proposed $HMM_{lstm}RL$	
		Precision (in%)	Recall (in%)	Precision (in%)	Recall (in %)
1	10000	68	58	65	63
2	20000	70	69	76	72.5
3	30000	81	76	87	86.5

Table 13: Comparison of various statistical machine translation system using BLEU score

S. No.	Methodology	Source (L)	Target (L)	BLEU Score
1	HMM [18]	English	Part of Speech	0.2287
2	Topic-based coherence model [21]	Indonesian	Japanese	0.1723
3	Proposed bi-directional HMM _{lstm} RL	Hindi	Tamil	0.7394
4	Proposed HMM _{lstm} RL	Hindi	Tamil	0.7637

5 Conclusion

In today's multicultural business world, there is an increased use of natural languages for establishing communication in the business environment. As a result of globalisation, machine translation systems are required to aid in communication between various organisations. Thus, the demand for translation systems into various language pairs has increased. It can also help in improving communication between people of different origins. This paper presented a HMM_{lstm}RL model HMM integrated with bi-directional LSTM with machine learning technique with the MapReduce framework for clustering. Finally, the proposed HMM_{lstm}RL model uses voting for language whether it belongs to a keyword or not. The analysis of the results showed that the proposed model exhibits higher performance than the existing techniques.

Acknowledgement: The authors would like to thank the research and development departments of B V Raju Institute of Technology, Vasavi College of Engineering, C.B.I.T and K.S.R.M College of Engineering for supporting this work.

Funding Statement: The author(s) received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

- [1] K. Chowdhary, "Natural language processing," in *Fundamentals of Artificial Intelligence*, New Delhi: Springer, pp. 603–649, 2020.
- [2] X. Qiu, T. Sun, Y. Xu, Y. Shao, N. Dai *et al.*, "Pre-trained models for natural language processing: A survey," *Science China Technological Sciences*, vol. 63, no. 10, pp. 1872–1897, 2020.
- [3] H. Xiong and R. Mamon, "An enabling framework for automated extraction of signals from market information in real time," *Knowledge-Based Systems*, vol. 246, pp. 108612, 2022.
- [4] D. W. Otter, J. R. Medina and J. K. Kalita, "A survey of the usages of deep learning for natural language processing," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 2, pp. 604–624, 2020.
- [5] R. Barzilay and K. McKeown, "Sentence fusion for multi document news summarization," *Computer Linguistic*, vol. 31, no. 3, pp. 297–328, 2005.
- [6] S. Zhong and J. Ghosh, "Generative model-based document clustering: A comparative study," *Knowledge and Information Systems*, vol. 8, no. 3, pp. 374–384, 2005.
- [7] U. Srilakshmi, S. A. Alghamdi, V. A. Vuyyuru, N. Veeraiah and Y. Alotaibi, "A secure optimization routing algorithm for mobile ad hoc networks," *IEEE Access*, vol. 10, pp. 14260–14269, 2022.

- [8] A. Siddharthan, A. Nenkova and K. McKeown, "Information status distinctions and referring expressions: An empirical study of references to people in news summaries," *Computer Linguistics*, vol. 37, no. 4, pp. 811–842, 2011.
- [9] J. Zhang, Y. Zhou and C. Zong, "Abstractive cross-language summarization via translation model enhanced predicate argument structure fusing," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 10, pp. 1842–1853, 2016.
- [10] D. Jiang, Y. Tong and Y. Song, "Cross-lingual topic discovery from multilingual search engine query log," *ACM Transactions on Information Systems*, vol. 35, no. 2, pp. 1–28, 2016.
- [11] I. Vulić, W. De Smet and M. F. Moens, "Cross-language information retrieval models based on latent topic models trained with document-aligned comparable corpora," *Information Retrieval*, vol. 16, no. 3, pp. 331–368, 2013.
- [12] S. Sra, "A short note on parameter approximation for von mises-fisher distributions: And a fast implementation of $I_s(x)$," *Computational Statistics*, vol. 27, no. 1, pp. 177–190, 2012.
- [13] A. Esuli, A. Moreo and F. Sebastiani, "Funnelling: A new ensemble method for heterogeneous transfer learning and its application to cross-lingual text classification," *ACM Transactions on Information Systems*, vol. 37, no. 3, pp. 1–30, 2019.
- [14] G. Heyman, I. Vulić and M. F. Moens, "C-Bilda extracting cross-lingual topics from non-parallel texts by distinguishing shared from unshared content," *Data Mining and Knowledge Discovery*, vol. 30, no. 5, pp. 1299–1323, 2016.
- [15] S. Hao and M. J. Paul, "An empirical study on cross lingual transfer in probabilistic topic models," *Computer Linguistics*, vol. 46, no. 1, pp. 1–40, 2020.
- [16] C. H. Chang and S. Y. Hwang, "A word embedding-based approach to cross-lingual topic modelling," *Knowledge and Information Systems*, vol. 63, no. 6, pp. 1529–1555, 2021.
- [17] K. V. Kumar and D. Yadav, "Word sense based hindi-tamil statistical machine translation," *International Journal of Intelligent Information Technologies*, vol. 14, no. 1, pp. 17–27, 2018.
- [18] I. Perikos, S. Kardakis and I. Hatzilygeroudis, "Sentiment analysis using novel and interpretable architectures of hidden markov models," *Knowledge-Based Systems*, vol. 229, pp. 107332, 2021.
- [19] E. L. Pontes, S. Huet, J. M. Torres-Moreno and A. C. Linhares, "Compressive approaches for cross-language multi-document summarization," *Data & Knowledge Engineering*, vol. 125, pp. 101763, 2020.
- [20] M. Cao and H. Zhuge, "Grouping sentences as better language unit for extractive text summarization," *Future Generation Computer Systems*, vol. 109, pp. 331–359, 2020.
- [21] D. Alahmadi, A. Wali and S. Alzahrani, "TAAM: Topic-aware abstractive arabic text summarisation using deep recurrent neural networks," *Journal of King Saud University-Computer and Information Sciences*, vol. 34, no. 6, pp. 2651–2665, 2022.
- [22] L. Li, J. Geissinger, W. A. Ingram and E. A. Fox, "Teaching natural language processing through big data text summarization with problem-based learning," *Data and Information Management*, vol. 4, no. 1, pp. 18–43, 2020.
- [23] C. Barros, E. Lloret, E. Saquete and B. Navarro-Colorado, "NATSUM: Narrative abstractive summarization through cross-document timeline generation," *Information Processing & Management*, vol. 56, no. 5, pp. 1775–1793, 2019.
- [24] S. Liu, L. Yang and X. Cai, "Syntax-enriched abstractive summarization," *Expert Systems with Applications*, vol. 199, pp. 116819, 2022.
- [25] A. McCallum, X. Wang and A. Corrada-Emmanuel, "Topic and role discovery in social networks with experiments on enron and academic email," *Journal of Artificial Intelligence Research*, vol. 30, pp. 249–272, 2007.
- [26] D. Q. Nguyen, R. Billingsley, L. Du and M. Johnson, "Improving topic models with latent feature word representations," *Transactions of the Association for Computational Linguistics*, vol. 3, pp. 299–313, 2015.
- [27] S. Ruder, I. Vulić and A. Søgaard, "A survey of cross-lingual word embedding models," *Journal of Artificial Intelligence Research*, vol. 65, no. 1, pp. 569–630, 2019.

- [28] N. Veeraiah, O. I. Khalaf, C. V. P. R. Prasad, Y. Alotaibi, A. Alsufyani *et al.*, “Trust aware secure energy efficient hybrid protocol for manet,” *IEEE Access*, vol. 9, pp. 120996–121005, 2021.
- [29] N. Veeraiah and B. T. Krishna, “Trust-aware fuzzyclus-fuzzy nb: Intrusion detection scheme based on fuzzy clustering and Bayesian rule,” *Wireless Networks*, vol. 25, pp. 4021–4035, 2019.
- [30] U. Srilakshmi, N. Veeraiah, Y. Alotaibi, S. A. Alghamdi, O. I. Khalaf *et al.*, “An improved hybrid secure multipath routing protocol for manet,” *IEEE Access*, vol. 9, pp. 163043–163053, 2021.
- [31] P. Kollapudi, S. Alghamdi, N. Veeraiah, Y. Alotaibi, S. Thotakura *et al.*, “A new method for scene classification from the remote sensing images,” *Computers, Materials & Continua*, vol. 72, no. 1, pp. 1339–1355, 2022.