CHAPTER2

APPROACHES AND LITERATURE SURVEY

The first section of this chapter is used to describe the different developments in Kannada language natural language processing. The remainder of this chapter gives a brief description about the different approaches and major developments in computational linguistic tools like Machine Transliteration, Parts of Tagger, Morphological Analyzer and Generator, Syntactic Parser and MT systems for different Indian languages.

2.1 LITERATURE SURVEY ON KANNADA NLP

The NLP though growing rapidly is still an immature area in Kannada language.Literature survey shows that, the development in natural language processing for Kannada is not explored much and is in beginning stage when compared to other Indian languages.There are very few developments found in Kannada NLP and some of these are under progress. The following are the major developments in Kannada NLP:

i) A computer program called Nudi was developed in 2004 [12]by the Kannada Ganaka Parishat. This font-encoding standard is used for managing and displaying the Kannada script. The government of Karnataka owns and makes the Nudi software free for use. Most of the Nudi fonts can be used for dynamic font embedding purposes as well as other situation like database management. Although Nudi is a font-encoding based standard which uses ASCII values to store glyphs, it provides inputting data in Unicode as well as saving data in Unicode. Nudi engine supports most of the window based database systems like Access, Oracle, SQL, DB2 etc. It also supports MySQL.

ii) Baraha software is a tool that functions as a phonetic keyboard for any Indian language including Kannada[13]. The first version of the Baraha was released in 1998 with an intention to provide free, friendly use Kannada language software, to enable even non-computer professionals to use Kannada in computers. Indirectly it aims to promote Kannada language in the cyber world. As a result millions of people across the world are now using Baraha for creating content in Indian languages. The main objective of the Baraha is "portability of data", so that Baraha can export the data in various data formats such as ANSI text, Unicode text, RTF, HTML.

iii) B.M. Sagar, Dr. ShobhaG and Dr. Ramakanth Kumar P proposed a work on Kannada Optical Character Recognition (OCR) in 2008 [14]. The process of converting the textual image into the machine editable format is called Optical Character Recognition. The main need for OCR arises in the context of digitizing the documents from the library, which helps in sharing the data through the Internet. The preprocessing, segmentation, Character Recognition and Post-processing are the four important modules in the proposed OCR system. Post processing technique uses a dictionary based approach implemented using Ternary Search Tree data structure which in turn increases the performance of the OCR output.

iv) T V Ashwin and P S Sastry developed a font and size-independent OCR system for printed Kannada documents using SVM in 2002 [15]. The input to the OCR system would be the scanned image of a page of text and the output is a machine editable file compatible with most typesetting software. At first the system extracts words from the document image and then segments the words into sub-character level pieces using a segmentation algorithm. In their work, they proposed a novel set of features for the recognition problem which are computationally simple to extract. A number of 2-class classifiers based on the SVM method was used for final recognition. The main characteristic is that, the proposed system is independent of the font and size of the printed text and the system is seen to deliver reasonable performance.

v) B.M. Sagar, Dr. ShobhaG and Dr. Ramakanth Kumar P proposed another work related to OCR for Kannada language in 2008 [16]. The proposed OCR system is used for the recognition of printed Kannada text, which can handle all types of Kannada characters. The system is based on database approach for character recognition. This system works in three levels in such a way that, first extracts image of Kannada scripts, then from the image to line segmentation and finally segments the words into sub-character level pieces. They reported that the level of accuracy of the proposed OCR system reached to 100%. The main limitation of this database approach is that for each character we need to have details like Character ASCII value, Character name, Character BMP image, Character width, length and total number of ON pixel in the image. Which in turn consumes more space as well as computationally complexity is high in recognizing the character.

vi) R Sanjeev Kunte and R D Sudhakar Samual proposed a simple and efficient optical character recognition system for basic symbols in printed Kannada text in 2007 [17]. The developed system recognizes basic characters such as vowels and consonants of printed Kannada text, which can handle different font sizes and font types. The system extracts the features of printed Kannada characters using Hu’s invariant moments and Zernike moments approach. The system effectively used Neural classifiers for the classification of characters based on moment features. The developer reported an encouraging recognition rate of 96·8%.

vii) A Kannada indexing software prototype is developed by Settar in 2002 [18]. This work deals with an efficient, user-friendly and reliable tool for automatic generation of index to Kannada documents. The proposed system is intended to benefit those who work on Kannada texts and is an improvement on any that exists in the languages. The input to the system may come either from an Optical Character Recognition system if it is made available, or from typeset documents. The output provides an editable and searchable index. Results indicate that the application is fast, comprehensive, effective and error free.

viii) A Kannada Wordnet was attempted by Sahoo and Vidyasagar of Indian Inst. of Technology. Bangalore, in 2003 [19]. Kannada WordNet serves as an on-line thesaurus andrepresents a very useful linguistic resource that helps in many NLP tasks such as MT, Information retrieval, word sense disambiguation,interface to internet search engines, text classification etc, in Kannada. The developed Kannada WordNet design has been inspired by the famous English WordNet, and to certain extent, by the Hindi WordNet. The most significant feature of WordNet is the semantic organization. The efficient underlying database design designed to handle storage and display of Kannada Unicode characters. The proposed WordNet would not only add to the sparse collection of machine-readable Kannada dictionaries, but also will give new insights into the Kannada vocabulary. It will provide sufficient interface for applications involved in Kannada MT, Spell Checker and Semantic Analyzer.

ix) In the year 2009 Amrita University, Coimbatore started to develop a Kannada WordNet project under the supervision of Dr K P Soman [20]. This NLP project is funded by Ministry of Human Resource and Management (MHRD) as a part of developing translation tools for Indian languages. A WordNet is a lexical database, with characteristics of both a dictionary and a thesaurus. This is an essential component of any MT System. The design of this online lexical reference system is inspired by current psycholinguistic and computational theories of human lexical memory. Nouns, verbs, adjectives and adverbs are organized into synonymous sets, each representing one underlying lexicalized concept. Different semantic relations link the synonyms sets. The most ambitious feature of a WordNet is the organization of lexical information in terms of word meanings rather than word forms.

x) T. N. Vikram and Shalini R Urs developed a prototype of morphological analyzer for Kannada language (2007) based on Finite State Machine [3]. This is just a prototype based on Finite state machines and can simultaneously serve as a stemmer, part of speech tagger and spell checker. The proposed morphological analyzer tool does not handle compound formation morphology and can handle a maximum of 500 distinct nouns and verbs.

xi) B.M. Sagar, Shobha G and Ramakanth Kumar P (2009) proposed a method for solving the Noun Phrase and Verb Phrase agreement in Kannada language sentences using CFG [21]. The system uses Recursive Descent Parser to parse the CFG and for given sentence parser identify the syntactic correctness of that sentence depending upon the Noun and Verb agreement. The system was tested with around 200 sample sentences.

xii) Uma Maheshwar Rao G. and Parameshwari K. of CALTS, University of Hyderabad attempted to develop a morphological analyzer and generators for South Dravidian languages in 2010 [22].

xiii) MORPH- A network and process model for Kannada morphological analysis/ generation was developed by K. Narayana Murthy and the performance of the system is 60 to 70% on general texts [23].

xiv) The University of Hyderabad under K. Narayana Murthy has worked on an English-Kannada MT system called “UCSG-based English-Kannada MT”, using the Universal Clause Structure Grammar (UCSG) formalism.

xv) Recently Shambhavi B. R and Dr. Ramakanth Kumar of RV College, Bangalore developed a paradigm based morphological generator and analyzer using a trie based data strucure [24]. The disadvantage of trie is that it consumes more memory as each node can have at most ‘y’ children, where y is the alphabet count of the language. As a result it can handle up to maximum 3700 root words and around 88K inflected words.

2.2 MACHINE TRANSLITERATION FOR INDIAN LANGUAGES

This section addresses the different developments in Indian language machine transliteration system, which is considered as a very important task needed for manyNLP applications. Machine transliteration is an important NLP tool required mainly for translating named entities from one language to another. Even though a number of different transliteration mechanisms are available to the world’s top level languages like English, European languages and Asian languages like Chinese, Japanese, Korean and Arabic. Still it is an initial stage for Indian languages. Literature shows that, recently some recognizable attempts have been done for few Indian languages like Hindi, Bengali, Telugu, Kannada and Tamil languages.

2.2.1 Major Contribution to Machine Transliteration

The Fig. 2.1 shows different researchers who contributed towards the developments of various machine transliteration systems.

The very first attempt in transliteration was done by Arababi through a combination of neural network and expert systems for transliterating from Arabic to English in 1994 [25]. The proposed neural network and knowledge-based hybrid system generate multiple English spellings for Arabic names.

The next development in transliteration was based on a statistical based approach proposed by Knight and Graehl in 1998 for back transliteration from English to Japanese and Katakana. This approach was adapted by Stalls and Knight for back transliteration from Arabic to English.

There were three different machine transliteration developments in the year 2000, from three separate research teams. Oh and Choi developed a phoneme based model using rule based approach incorporating phonetics as an intermediate representation. This English-Korean (E-K) transliteration model is built using pronunciation and contextual rules. Kang, B. J. and K. S. Choi, in their work, presented an automatic character alignment method between English word and Korean transliteration. Aligned data is trained using supervised learning decision tree method to automatically induce transliteration and back-transliteration rules. This methodology is fully bi-directional, i.e. the same methodology is used for both transliteration and back transliteration. SungYoung Jung proposed a statistical English-to-Korean transliteration model that exploits various information sources. This model is a generalized model from a conventional statistical tagging model by extending Markov window with some mathematical approximation techniques. An alignment and syllabification method is developed for accurate and fast operation.

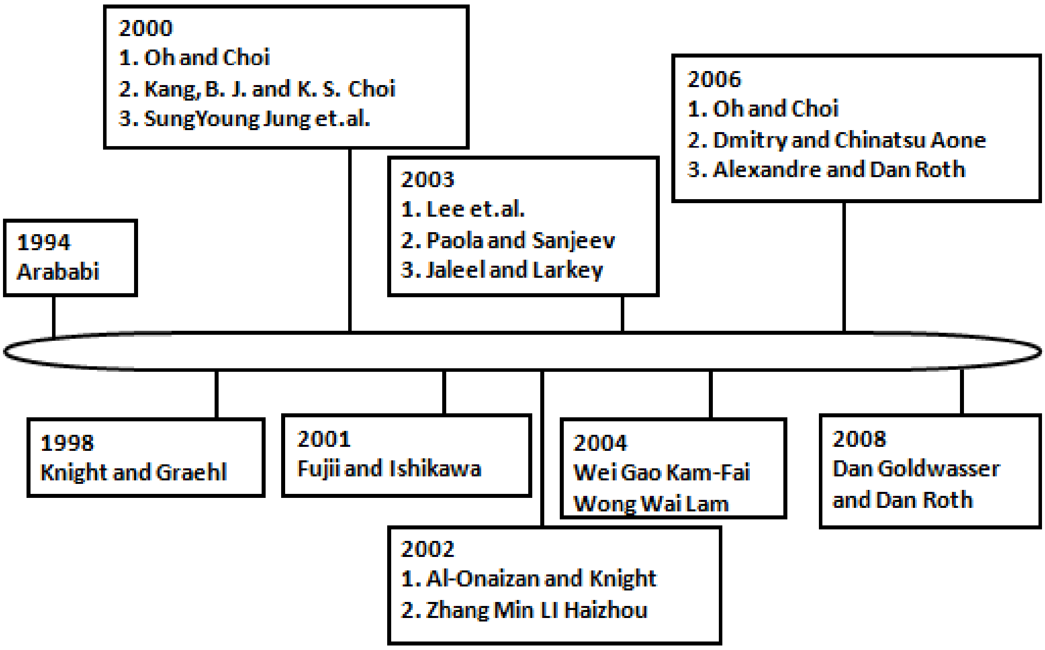

Fig.2.1: Contributors to Machine Transliteration

In the year 2001, Fujii and Ishikawa describe a transliteration system for English-Japanese Cross Lingual Information Retrieval (CLIR) task that requires linguistic knowledge.

In the year 2002, Al-Onaizan and Knight developed a hybrid model based on phonetic and spelling mappings using Finite state machines. The model was designed for transliterating Arabic names into English. In the same year, Zhang Min LI Haizhou SU Jianproposed a direct orthographic mapping framework to model phonetic equivalent association by fully exploring the orthographical contextual information and the orthographical mapping. Under the DOM framework, a joint source-channel transliteration model (*n*-gram TM) captures the source-target word orthographical mapping relation and the contextual information.

An English-Arabic transliteration scheme was developed by Jaleel and Larkey based on HMM using GIZA++ approach in 2003. Mean while they also attempted to develop a transliteration system for Indian language. Lee et.al. [2003]developed the noisy channel model for English Chinese language pair, in which the back transliteration problem is solved by finding the most probable word E, given transliteration C. Letting $P(E)$ be the probability of a word E, then for a given transliteration C, the back-transliteration probability of a word E can be written as $P(E|C)$. This method requires no conversion of source words into phonetic symbols. The model is trained automatically on a bilingual proper name list via unsupervised learning. Model parameters are estimated using EM. Then the channel decoder with Viterbi decoding algorithm is used to find the word $\hat{E}$,that is, the most likely to the word E that gives rise to the transliteration C. The model is tested for English Chinese language pair. In the same year Paola Virga and Sanjeev Khudanpur demonstrated the application of statistical machine translation techniques to “translate” the phonemic representation of an English name intoChinese. In this case transliteration is obtained by using an automatic text-to-speech system, to a sequence of initials and finals.

Wei Gao Kam-Fai Wong Wai Lam proposed an efficient algorithm for phoneme alignment in 2004. In this a data driven technique is proposed for transliterating English names to their Chinese counterparts, i.e. forward transliteration. With the same set of statistics and algorithms, transformation knowledge is acquired automatically by machine learning from existing origin-transliteration name pairs, irrespective of specific dialectal features implied. The method starts off with direct estimation for transliteration model, which is then combined with target language model for postprocessing of generated transliterations. Expectation-maximization (EM) algorithm is applied to find the best alignment (Viterbi alignment) for each training pair and generate symbol-mapping probabilities. A weighted finite state transducer (WFST) is built based on symbol-mapping probabilities, for the transcription of an input English phoneme sequence into its possible pinyin symbol sequences.

Dmitry Zelenko and Chinatsu Aoneproposed two discriminative methods for name transliteration in 2006. The methods correspond to local and global modelling approaches in modelling structured output spaces. Both methods do not require alignment of names in different languages but their features are computed directly from the names themselves. The methods are applied to name transliteration from three languages - Arabic, Korean and Russian into English. In the same year Alexandre Klementiev and Dan Roth developed a discriminativeapproach for transliteration. A linear model is trained to decide whether a word T isa transliteration of a NE S.

2.2.2 Machine Transliteration Approaches

Transliteration is generally classified into three types namely, Grapheme based, Phoneme based and hybrid models and correspondence-based transliteration model [26, 27]. These models are classified in terms of the units to be transliterated. The grapheme based approach (Lee & Choi, 1998; Jeong, Myaeng, Lee, & Choi, 1999; Kim, Lee, & Choi, 1999; Lee, 1999; Kang & Choi, 2000; Kang & Kim, 2000; Kang, 2001; Goto, Kato, Uratani, & Ehara, 2003; Li, Zhang, & Su, 2004) treat transliteration as an orthographic process and tries to map the source graphemes directly to the target graphemes. Grapheme based model is further divided into (i) source channel model (ii) Maximum Entropy Model (iii) Conditional Random Field models and (iv) Decision Trees model. The grapheme-based transliteration model is sometimes referred to as the direct method because it directly transforms source language graphemes into target language graphemes without any phonetic knowledge of the source language words.

On the other hand, phoneme based models (Knight & Graehl, 1997; Lee, 1999; Jung, Hong, & Paek, 2000; Meng, Lo, Chen, & Tang, 2001) treat transliteration as a phonetic process rather than an orthographic process. WFST and extended Markov window (EMW) are the approaches belonging to the phoneme based models. The phoneme-based transliteration model is sometimes referred to as the pivotal methodbecause it uses source language phonemes as a pivot when it produces target language graphemes from source language graphemes. This modeltherefore usually needs two steps: 1) produce source language phonemes from source language graphemes and 2) produce target language graphemes from source phonemes.

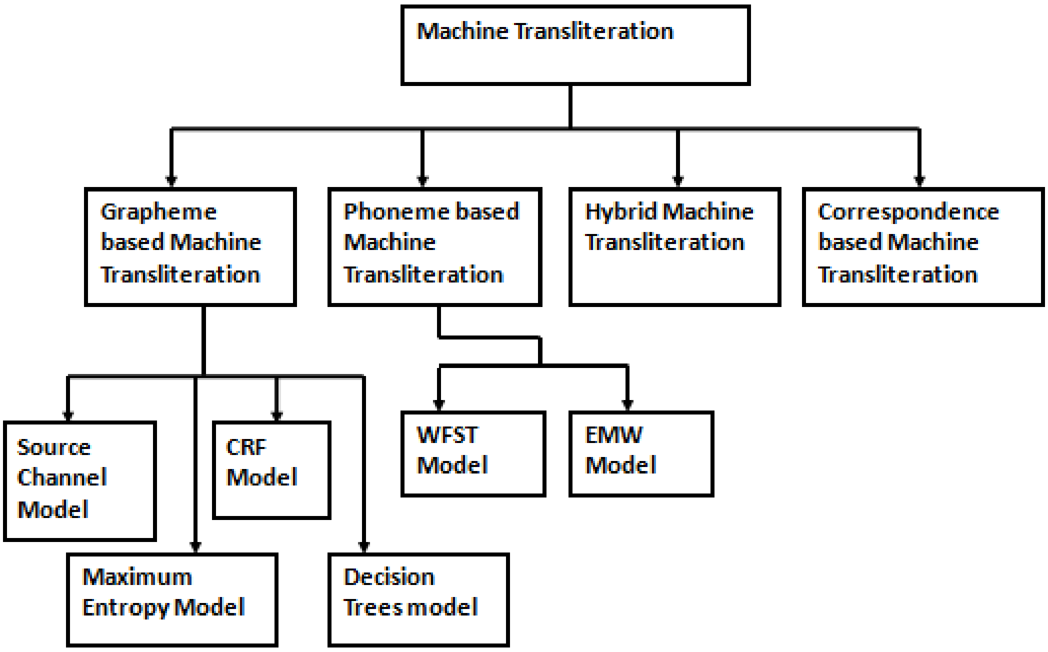
Machine Transliteration
Grapheme based Machine Transliteration
Phoneme based Machine Transliteration
Hybrid Machine Transliteration
Correspondence based Machine Transliteration
Source Channel Model
CRF Model
Maximum Entropy Model
Decision Trees model
WFST Model
EMW Model

Fig. 2.2: General Classification of Machine Transliteration System

As the name indicates, a hybrid model (Lee, 1999; Al-Onaizan & Knight, 2002; Bilac & Tanaka, 2004) either use a combination of a grapheme based model and a phoneme based model or capture the correspondence between source graphemes and source phonemes to produce target language graphemes. Correspondence-based transliteration model was proposed by Oh & Choi, in the year 2002. The hybrid transliteration modeland correspondence-based transliteration modelmake use of both source language graphemes and source language phonemes while producing target language transliterations. Fig. 2.2 shows the general classification of machine transliteration system.

2.2.3 Machine Transliteration in India: A Literature Survey

2.2.3.1 English to Hindi Machine Transliteration

Literature shows that majority of work in machine transliteration for Indian languages were done in Hindi and Dravidian languages. The following are the noticeable developments in English to Hindi or other Indian languages to Hindi machine transliteration.

i) Transliteration as a Phrase Based Statistical MT: In 2009, Taraka Rama and Karthik Gali addressed the transliteration problem as translation problem [27]. They have used the popular phrase based SMT systems successfully for the task of transliteration. This is a stochastic based approach, where the publicly available GIZA++ and beam search based decoder were used for developing the transliteration model. A well organized English- Hindi aligned corpus was used to train and test the system. It was a prototype system and reported an accuracy of 46.3% on the test set.

ii) Another transliteration system was developed by Amitava Das, Asif Ekbal, Tapabrata Mandal and Sivaji Bandyopadhyay based on NEWS 2009 Machine Transliteration Shared Task training datasets [26]. The proposed transliteration system uses the modified joint source channel model along with two other alternatives to translate English to Hindi transliteration. The system also uses some post processing rules for the purpose of removing the errors in the system to improve the accuracy. They performed one standard run and two nonstandard runs in the developed English to Hindi transliteration system. The results showed that the performance of the standard run was better than the non standard one.

iii) Using the Letter- to- Phoneme technology, the transliteration problem was addressed by Amitava Das, Asif Ekbal, Tapabrata Mandal and Sivaji Bandyopadhyay in 2009 [26]. This approach was intended for improving the performance of the existing work with re-implementation using the specified technology. In the proposed system, transliteration problem is interpreted as a variant of the Letter-to-Phoneme (L2P) subtask of text to- speech processing. They apply a re-implementation of a state-of-the-art, discriminative L2P system to the problem, without further modification. In their experiment, they demonstrated that an automatic letter-to- phoneme transducer performs fairly well with no language specific or transliteration-specific modifications.

iv) An English to Hindi Transliteration using Context-Informed Phrase Based Statistical Machine Translation (PBSMT) was proposed by Rejwanul Haque, Sandipan Dandapat, Ankit Kumar Srivastava, Sudip Kumar Naskar and Andy Way CNGL in 2009 [26]. The transliteration system was modelled by translating characters rather than words as in character-level translation systems. They used a memory-based classification framework that enables efficient estimation of these features while avoiding data sparseness problems. The experiments were both at character and Transliteration Unit (TU) level and reported that position - dependent source context features produce significant improvements in terms of all evaluation metrics. In this way the problem of machine transliteration was successfully implemented by adding source context modelling into state-of-the-art log-linear PB-SMT. In their experiment, they also showed that, by taking source context into account, they can improve the system performance substantially.

v) Abbas Malik, Laurent Besacier Christian Boitet and Pushpak Bhattacharyya proposed an Urdu to Hindi Transliteration using hybrid approach in 2009 [26]. This hybrid approach combines Finite State Machine (FSM) based techniques with statistical word language model based approach and achieved better performance. The main effort of this system was to remove diacritical marks from the input Urdu text. They reported that the approach improved the system accuracy by 28.3% in comparison with their previous finite-state transliteration model.

vi) A Punjabi to Hindi transliteration system was developed by Gurpreet Singh Josan and Jagroop Kaur, based on statistical approach in 2011 [25]. The system used letter to letter mapping as baseline and tried to find out the improvements by statistical methods. They used a Punjabi – Hindi parallel corpus for training and publicly available SMT tools for building the system.

2.2.3.2 English to Tamil Language Machine Transliteration

The first English to Tamil transliteration system was developed by Kumaran A and Tobias Kellner in the year 2007. Afraz and Sobha developed a statistical transliteration system using statistical approach in the year 2008. The third transliteration system was based on Compressed Word Format (CWF) algorithm and a modified version of Levenshtein’s Edit Distance algorithm. Vijaya MS, Ajith VP, Shivapratap G and Soman KP of Amrita University, Coimbatore proposed the remaining three English to Tamil Transliteration using different approaches.

i) Kumaran A and Tobias Kellner proposed machine transliteration framework based on a core algorithm modelled as a noisy channel, where the source string gets garbledinto target string. Viterbi alignment was used for source and target language segments alignment. The transliteration is learned by estimating the parameters of the distribution that maximizes the likelihood of observing the garbling seen in the training data using Expectation Maximization algorithm. Subsequently, given a target language string ‘t’, the most probable source language string ‘s’that gave raise to ‘t’, is decoded. The method is applied for forward transliteration from English to Hindi, Tamil, Arabic, Japanese and backward transliteration from Hindi, Tamil, Arabic, and Japanese to English.

ii) Afraz and Sobha developed a statistical transliteration engine using an n-grams based approach in the year 2008. This algorithm uses n-gram frequencies of the transliteration units, to find the probabilities. Each transliteration unit is pattern of consonant-vowel in the word. This transliteration engine is used in their Tamil to English CLIR system.

iii) Srinivasan C Janarthanam et.al. (2008)proposed an efficient algorithm for transliteration of English named entities to Tamil. In the first stage of transliteration process, he used a Compressed Word Format (CWF) algorithm to compress both English and Tamil named entities from their actual forms. Compressed Word Format of words is created using an ordered set of rewrite and remove rules. Rewrite rules replace characters and clusters of characters with other characters or clusters. Remove rules simply remove the characters or clusters. This CWF algorithm is used for both English and Tamil names, but with different rule set. The final CWF forms will only have the minimal consonant skeleton. In the second stage Levenshtein’s Edit Distance algorithm is modified to incorporate Tamil characteristics like long-short vowel, ambiguities in consonants like ‘n’, ‘r’, ‘i’, etc. Finally, the CWF Mapping transliteration algorithm takes an input source language named entity string, converts it into CWF form and then maps with similar Tamil CWF words using modified edit distance. This method produces a ranked list of transliterated names in the target language Tamil for an English source language name.

iv) In the first attempt,Vijaya MS and colleagues demonstrated a transliteration model for English to Tamil transliteration using Memory based learning by reformulating the transliteration problem as sequence labelling and multi classification in 2008 [28]. The proposed system was corpus based and they have used English- Tamil aligned parallel corpus of 30,000 person names and 30,000 place names to train the transliteration model. They evaluated the performance of the system based on top 5 accuracy and reported 84.16% exact English to Tamil transliteration.

v) In their second attempt, the transliteration problem was modelled as classification problem and trained using C4.5 decision tree classifier, in WEKA Environment [29]. The same parallel corpus was used to extract features and these features were used to train the WEKA algorithm. The resultant rules generated by the WEKA were used to develop the transliteration system. They reported exact Tamil transliterations for 84.82% of English names.

vi) The third English to Tamil Transliteration was developed using One Class Support Vector Machine algorithm in 2010 [30]. This is a statistical based transliteration system, where training, testing and evaluations were performed with publicly available SVM tool. The experiment result shows that, the SVM based transliteration was outperformed over other previous methods.

2.2.3.3 English to Malayalam Language Machine Transliteration

In the year 2009, Sumaja Sasidharan, Loganathan R, and Soman K P developed Englishto MalayalamTransliteration using Sequence labelling approach [31]. They have used a parallel corps consisting of 20000 aligned English-Malayalam person names for training the system. The approach is very similar to earlier English to Tamil transliteration. The model produced the Malayalam transliteration of English words with an accuracy of 90% when tested with 1000 names.

2.2.3.4 English to Telugu Language Machine Transliteration

An application of transliteration was proposed by V.B. Sowmya and Vasudeva Varmain in 2009 [32]. They proposed a transliteration based text input method for Telugu language using simple edit-distance based approach. The user type Telugu using Roman script. They have tested the approach with three datasets – general data, countries name and place-person names and reported the performance of the system.

2.2.3.5 English to Indian Language Machine Transliteration

A well known online transliteration system for Indian language is Google Indic transliteration which works reasonably well for English to Indian languages. There are also Keyboard layouts like Inscript and Keylekh transliteration that have been available for Indian languages. The following are the generic approaches for machine transliteration for English to Indian languages.

i) Harshit Surana and Anil Kumar Singh in 2008, proposed a transliteration system using two different methods on two Indian languages Hindi and Telugu [33]. In their experiment, using character based n-grams, a word is classified into two classes, either Indian or foreign. The proposed technique considered the properties of the scripts but does not require any training data on the target side, while it uses more sophisticated techniques on the source side. The proposed model first identifies the class of the source side word to identify whether it is a foreign or Indian word. Based on the identified class, the system uses any one of the two methods. The system uses the easily creatable mapping tables and a fuzzy string matching algorithm to get the target word.

ii) Amitava Das, Asif Ekbal, Tapabrata Mandal and Sivaji Bandyopadhyay proposed a transliteration technique based on orthographic rules and phoneme based approach and system was trained on the NEWS 2010 transliteration datasets [34]. In their experiments, one standard run and two non-standard runs were submitted for English to Hindi and Bengali transliteration while one standard and one non-standard run were submitted for Kannada and Tamil. The reported results were as follows: For the standard run, the system demonstrated means F-Score values of 0.818 for Bengali, 0.714 for Hindi, 0.663 for Kannada and 0.563 for Tamil. The reported mean F-Score values of non-standard runs are 0.845 and 0.875 for Bengali non-standard run-1 and 2, 0.752 and 0.739 for Hindi non-standard run-1 and 2, 0.662 for Kannada non-standard run-1 and 0.760 for Tamil non-standard run-1. Non-Standard Run-2 for Bengali has achieved the highest score amongst all the submitted runs. Hindi Non-Standard Run-1 and Run-2 runs are ranked as the 5th and 6th among all submitted Runs.

iii) K Saravaran, Raghavendra Udupa and A Kumaran proposed a CLIR system enhanced with transliteration generation and mining in 2010 [35]. They proposed Hindi-English and Tamil-English cross-lingual evaluation tasks, in addition to the English-English monolingual task. They used a language modelling based approach using query likelihood based document ranking and a probabilistic translation lexicon learned from English-Hindi and English-Tamil parallel corpora. To deal with out-of-vocabulary terms in the cross-lingual runs, they proposed two specific techniques. The first technique is to generate transliterations directly or transitively, and second technique is to mining possible transliteration equivalents from the documents retrieved in the first-pass. In their experiment they showed that both of these techniques significantly improved the overall retrieval performance of our cross-lingual IR system. The systems achieved a peak performance of a MAP of 0.4977 in Hindi-English and 0.4145 in the Tamil-English.

iv) DLI developed a unified representation for Indian languages called an Om transliteration which is similar to ITRANS (Indian language Transliteration Scheme) [36]. To enhance the usability and readability, Om has been designed on the following principles: (i) easy readability (ii) case-insensitive mapping and (iii) phonetic mapping, as much as possible. In Om transliteration system, when a user is not interested in installing language components, or when the user cannot read native language scripts the text may be read in English transliteration itself. Even in the absence of Om to native font converters, people around the globe can type and publish texts in the Om scheme which can be read and understood by many, even when they cannot read the native script.

v) Using statistical alignment models and Conditional Random Fields (CRF), a language independent transliteration system was developed by Shishtla, Surya Ganesh V, Sethuramalingam Subramaniam and Vasudeva Varma in 2009 [26]. Using the expectation maximization algorithm, statistical alignment models maximizes the probability of the observed (source, target) word pairs and then the character level alignments are set to maximum posterior predictions of the model. The advantage of the system is that no language-specific heuristics were used in any of the modules and hence it is extensible to any language-pair with least effort.

vi) Using PBSMT approach, English-Hindi, English-Tamil and English-Kannada transliteration systems were developed by Manoj Kumar Chinnakotla and Om P. Damani in 2009 [26]. In the proposed SMT based system, words are replaced by characters and sentences by words and GIZA++ was used for learning alignments and Moses for learning the phrase tables and decoding. In addition to standard SMT parameters tuning, the system also focus on tuning the Character Sequence Model (CSM) related parameters like order of the CSM, weight assigned to CSM during decoding and corpus used for CSM estimation. The results show that improving the accuracy of CSM pays off in terms of improved transliteration accuracies.

vii) Kommaluri Vijayanand, Inampudi Ramesh Babu and Poonguzhali Sandiran proposed the transliteration systems for English to Tamil language based on the reference corpora which consisted of language pair of 1000 names in 2009 [26]. The proposed transliteration system was implemented using JDK 1.6.0 for transliterating the English Named Entities into Tamil language. From the experiment they found that the accuracy in top-1 score of the system was 0.061.

viii) Transliteration between Indian languages and English using an EM algorithm was proposed by Dipankar Bose and Sudeshna Sarkar in 2009 [26]. They used an EM algorithm to learn the alignment between the languages. They found that there is lot of ambiguities in the rules mapping the characters in the source language to the corresponding characters in the target language. They handled some of these ambiguities by capturing context by learning multi-character based alignments and use of character n-gram models. They have used multiple models and a classifier to decide which model to use in their system. Both the models and classifiers are learned in a completely unsupervised manner. The performance of the system was tested for English and several Indian languages. They have used an additional preprocessor for Indian languages, which enhances the performance of the transliteration model. One more advantage is that, the proposed system is robust in the sense that it can filter out noise in the training corpus, can handle words of different origins by classifying them into different classes.

ix) Using word-origin detection and lexicon lookup method, an improvement in transliteration was proposed by Mitesh M. Khapra and Pushpak Bhattacharyya in 2009 [26]. The proposed improved model uses the following framework: (i) a word-origin detection engine (*pre-processing*) (ii) a CRF based transliteration engine and (iii) a re-ranking model based on lexicon lookup (*post-processing*). They applied their idea on *English-Hindi* and *English- Kannada* transliteration and reported 7.1% improvement in top-1 accuracy. The performance of the system was tested against the NEWS 2009 dataset. They submitted one standard run and one non-standard run for the English-Hindi task and one standard run for the English-Kannada task.

x) Sravana Reddy and Sonjia Waxmonsky proposed a substring-based transliteration with conditional random Fields for English to Hindi, Kannada and Tamil languages in 2009 [26]. The proposed transliteration system was based on the idea of phrase-based machine translation. In the transliteration system, phrases correspond to multi-character substrings. So, source and target language strings are treated not as sequences of characters but as sequences of non-overlapping substrings in the proposed system. Using CRFs, they modelled the transliteration as a 'sequential labelling task'where substring tokens in the source language are labelled with tokens in the target language. The system uses both 'local contexts'and 'phonemic information' acquired from an English pronunciation dictionary. They evaluated the performance of the system separately for Hindi, Kannada and Tamil languages using a CRF trained on the training and development data, with the feature set U+B+T+P.

xi) Balakrishnan Vardarajan and Delip Rao proposed an ε-extension Hidden Markov Model (HMM)'s and Weighted Transducers for Machine Transliteration from English to five different languages, including Tamil, Hindi, Russian, Chinese, and Kannada in 2009 [26]. The developed method involves deriving substring alignments from the training data and learning a weighted FST from these alignments. They have defined a ε -extension HMM to derive alignments between training pairs and a heuristic to extract the substring alignments. The performance of the transliteration system was evaluated based on the standard track data provided by the NEWS 2009. The main advantage of the proposed approach is that the system is language agnostic and can be trained for any language pair within a few minutes on a single core desktop computer.

xii) Raghavendra Udupa, K Saravanan, A Kumaran and Jagadeesh Jagarlamudi addressed the problem of mining transliterations of Named Entities (NEs) from large comparable corpora in 2009 [26]. They have proposed a mining algorithm called Mining Named-entity Transliteration equivalents (MINT), which uses a cross-language document similarity model to align multilingual news articles and then mines NETEs from the aligned articles using a transliteration similarity model. The main advantage of MINT is that, it addresses several challenges in mining NETEs from large comparable corpora: exhaustiveness (in mining sparse NETEs), computational efficiency (in scaling on corpora size), language independence (in being applicable to many language pairs) and linguistic frugality (in requiring minimal external linguistic resources). In their experiment they showed that the performance of the proposed method was significantly better than a state-of-the-art baseline and scaled to large comparable corpora.

xiii) Rohit Gupta, Pulkit Goyal and Sapan Diwakar proposed a transliteration system among Indian languages using WX Notation in 2010 [37]. They have proposed a new transliteration algorithm which is based on Unicode transformation format of an Indian language. They tested the performance of the proposed system on a large corpus having approximately 240k words in Hindi to other Indian languages. The accuracy of the system is based on the phonetic pronunciations of the words in target and source language and this was obtained from Linguistics having knowledge of both the languages. From the experiment, they found that the time efficiency of the system is better and it takes less than 0.100 seconds for transliterating 100 Devanagari (Hindi) words into Malayalam when run on an Intel Core 2 Duo, 1.8 GHz machine in Fedora.

xiv) A grapheme-based model was proposed by Janarthanam, Sethuramalingam and Nallasamy in 2008 [26]. In this proposed system, the transliteration equivalents are identified by matching in a target language database based on edit distance algorithm. The transliteration system was trained with several names and then the trained model is used to transliterate new names.

xv) In a separate attempt, Surana and Singh proposed another algorithm for transliteration in 2008 that eliminates the training phase by using fuzzy string matching approach [26].

2.3 PARTS OF SPEECH TAGGING FOR INDIAN LANGUAGES

Some classic examples for POS Taggers available in English are Bril tagger, Tree tagger, CLAWS tagger and online tagger ENGTWOL. In Indian languages, most of the natural language processing work has been done in Hindi, Tamil, Malayalam and Marathi. These languages have several part-of-speech taggers based on different mechanisms. Research on part-of-speech tagging has been closely tied to corpus linguistics. The Fig. 2.3 shows the development of various corpus and POS taggers using different approaches.

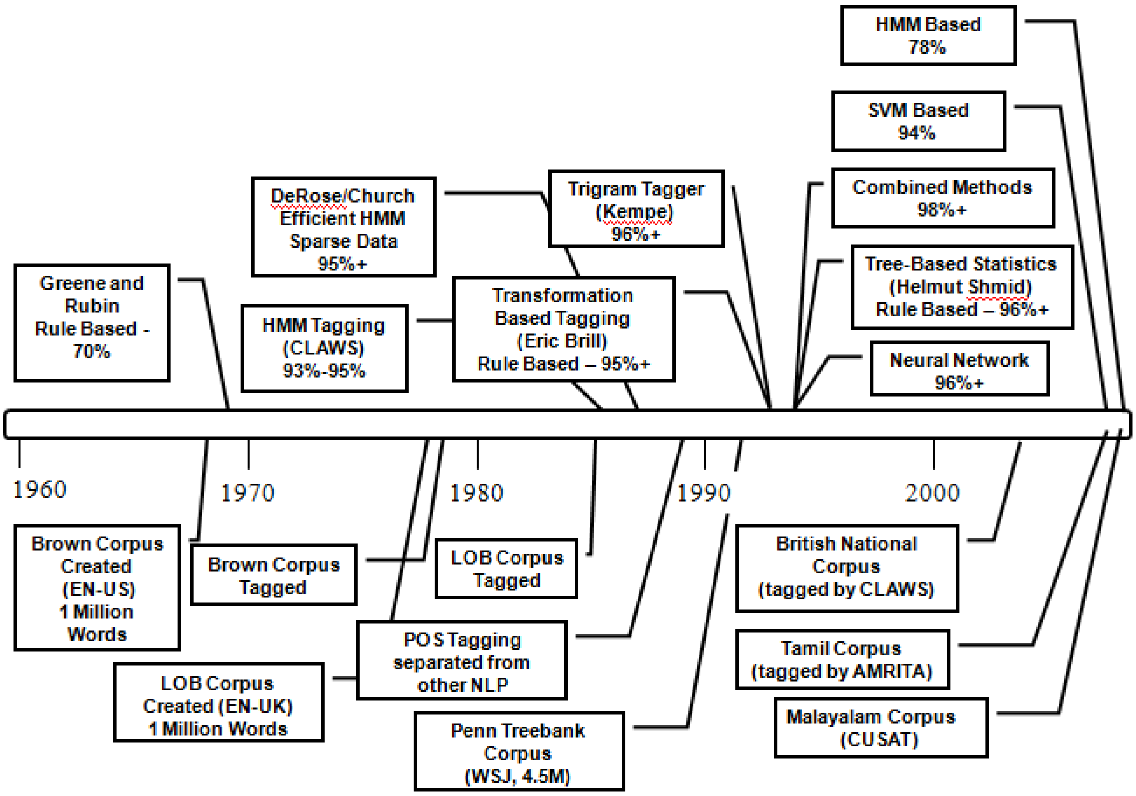
HMM Based 78%
SVM Based 94%
Combined Methods 98%+
Tree-Based Statistics (Helmut Shmid) Rule Based – 96%+
Neural Network 96%+
DeRose/Church Efficient HMM Sparse Data 95%+
Trigram Tagger (Kempe) 96%+
Greene and Rubin Rule Based - 70%
HMM Tagging (CLAWS) 93%-95%
Transformation Based Tagging (Eric Brill) Rule Based – 95%+
1960
1970
1980
1990
2000
Brown Corpus Created (EN-US) 1 Million Words
Brown Corpus Tagged
LOB Corpus Tagged
British National Corpus (tagged by CLAWS)
Tamil Corpus (tagged by AMRITA)
Malayalam Corpus (CUSAT)
POS Tagging separated from other NLP
LOB Corpus Created (EN-UK) 1 Million Words
Penn Treebank Corpus (WSJ, 4.5M)

Fig.2.3: Various Corpus and POS taggers

Earlier work in POS tagging for Indian languages was mainly based on rule based approaches. But the fact that rule-based method requires expert linguistic knowledge and hand written rule. Due to the morphological richness of Indian languages, researchers faced a great difficulty to write complex linguistic rules and the rule based approach did not result well in many cases. Later, researchers shifted to stochastic and other approaches and developed some better POS taggers in various Indian languages. Even though stochastic methods need very large corpora to be effective, many successful POS taggers were developed and used in various NLP tasks for Indian language.

In most of the Indian languages, the ambiguity is the key issue that must be addressed and solved while designing a POS tagger. For different context, words behave differently and hence the challenge is to correctly identify the POS tag of a token appearing in a particular context. Literature survey shows that, for Indian languages, POS taggers were developed only in Hindi, Bengali, Panjabi and Dravidian languages. Some noticeable attempts were done in Dravidian languages like Tamil, Telugu and Malayalam except Kannada language. Some POS taggers were also developed generic to the Hindi, Bengali and Telugu languages. All proposed POS taggers were based on various Tagset, developed by different organization and individuals. This section gives a survey on developments of various POS taggers in Indian languages. The following sub sections are organized as follows: The first sub section gives a brief description about various attempts in POS taggers in Indian languages. The second sub section is about the different Tagset developed for Indian languages.

2.3.1Parts of Speech Tagging Approaches

POS taggers are broadly classified into three categories called Rule based, Empirical based and Hybrid based .In case of rule based approach, hand-written rules are used to distinguish the tag ambiguity. The empirical POS taggers are further classified into Example based and Stochastic based taggers. Stochastic taggers are either HMM based, choosing the tag sequence which maximizes the product of word likelihood and tag sequence probability, or cue-based, using decision trees or maximum entropy models to combine probabilistic features. The stochastic taggers are further classified into supervised and unsupervised taggers. Each of these supervised and unsupervised taggers are categorized into different groups based on the particular algorithm used. The Fig.2.4 below shows the classification of parts of speech approaches.

In the recent literature, several approaches to POS tagging based on statistical and machine learning techniques are applied, including: HMMs, Maximum Entropy taggers, Transformation–based learning, Memory–based learning, Decision Trees, AdaBoost, and Support Vector Machines. Most of the taggers have been evaluated on the English WSJ corpus, using the Penn Treebank set of POS categories and a lexicon constructed directly from the annotated corpus. Although the evaluations were performed with slight variations, there was a wide consensus in the late 90’s that the state–of-the–art accuracy for English POS tagging was between 96.4% and 96.7%. In the recent years, the most successful and popular taggers in the NLP community have been the HMM–based TnT tagger, the Transformation Based Learning (TBL) tagger and several variants of the Maximum Entropy (ME) approach.

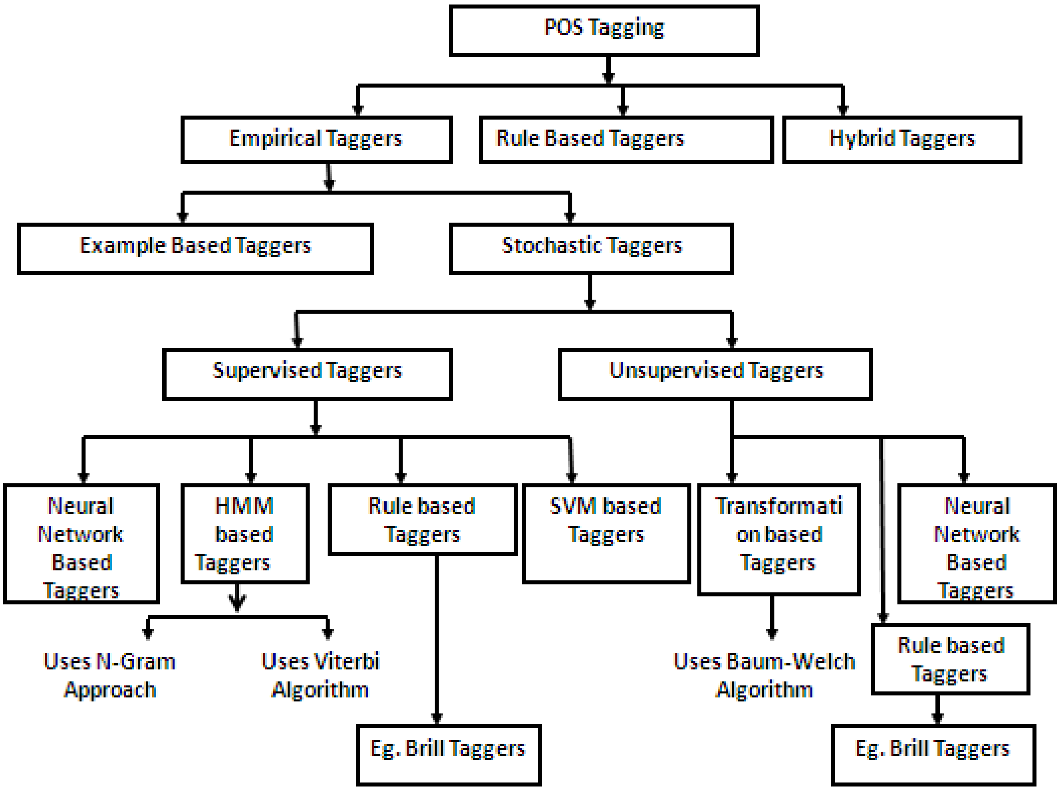

Fig.2.4: Classification of POS tagging Approaches

2.3.1.1 Rule Based POS tagging

The rule based POS tagging models apply a set of hand written rules and use contextual information to assign POS tags to words. These rules are often known as context frame rules. For example, a context frame rule might say something like: *"If an ambiguous/unknown word X is preceded by a Determiner and followed by a Noun, tag it as an Adjective"*. Brill's tagger is one of the first and widely used English POS Tagger that employs rule based algorithms.

The earliest algorithms for automatically assigning part-of-speech were based on two-stage architecture. The first stage used a dictionary to assign each word a list of potential parts of speech. The second stage used large lists of hand-written disambiguation rules to bring down this list to a single part-of-speech for each word. The ENGTWOL tagger is based on the same two-stage architecture, although both the lexicon and the disambiguation rules are much more sophisticated than the earlyalgorithms.

2.3.1.2 Empirical Based POS tagging

The relative failure of rule-based approaches, the increasing availability of machine readable text and the increase in capability of hardware with decrease in cost are some of the reasons for researchers to prefer corpus based POS tagging. The empirical approach of parts speech tagging is further divided into two categories: Example-based approach and Stochastic based approach. Literature shows that majority of the developed POS taggers belongs to empirical based approach.

2.3.1.2.1 Example-Based techniques

Example-Based techniques usually work in two steps. In the first step, it finds the training instance that is most similar to the current problem instance. In the next step, it assumes the same class for the new problem instance as for the similar one.

2.3.1.2.2 Stochastic based POS tagging

The stochastic approach explores the most frequently used tag for a specific word in the annotated training data and uses this information to tag that word in the annotated text. A stochastic approach required a sufficiently large sized corpus and calculates frequency, probability or statistics of each and every word in the corpus. The problem with this approach is that, it can come up with sequences of tags for sentences that are not acceptable according to the grammar rules of a language.

The use of probabilities in tags is quite old; probabilities in tagging were first used in 1965, a complete probabilistic tagger with Viterbi decoding was sketched by Bahl and Mercer (1976), and various stochastic taggers were built in the 1980's (Marshall, 1983; Garside, 1987; Church, 1988; DeRose, 1988).

2.3.1.2.2.1Supervised POS Tagging

The supervised POS tagging models require pre-tagged corpora which are used for training to learn information about the tagset, word-tag frequencies, rule-sets etc. The performance of the models generally increases with the increase in size of this corpus.

HMM based POS tagging: An alternative to word frequency approach is known as the n-gram approach, whereprobability of given sequence of tags are calculated. It determines the best tag for a word by calculating the probability that it occurs with the n previous tags, where the value of n is set to 1, 2 or 3 for practical purposes. These are known as the Unigram, Bigram and Trigram models. The most common algorithm for implementing the n-gram approach for tagging new text is known as the HMM's Viterbi Algorithm. The Viterbi algorithm is a search algorithm that avoids the polynomial expansion of a breadth first search by trimming the search tree at each level using the best 'm' Maximum Likelihood Estimates (MLE) where 'm' represents the number of tags of the following word. For a given sentence or word sequence, HMM taggers choose the tag sequence that maximizes the following formula:

$$P(\text{word} \mid \text{tag}) \text{ X } P(\text{tag} \mid \text{previous n tags}) \quad (2.1)$$

A bigram-HMM tagger of this kind chooses the tag t_i for word w_i that is most probable given the previous tag t_{i-1} and the current word w_i:

$$t_i = \arg\max_j P(t_j \mid t_{i-1}, w_i) \qquad (2.2)$$

Support Vector Machines: This is the powerful machine learning method used for various applications in NLP and other areas like bio-informatics. SVM is a machine learning algorithm for binary classification, which has been successfully applied to a number of practical problems, including NLP. Let $\{(x_1, y_1). . . (x_N, y_N)\}$ be the set of N training examples, where each instance x_i is a vector in $\mathbf{R}^N$and $y_i \in \{-1,+1\}$ is the class label. In their basic form, a SVM learns a linear hyperplane, that separates the set of positive examples from the set of negative examples with maximal margin (the margin is defined as the distance of the hyperplane to the nearest of the positive and negative examples). This learning bias has proved to good in terms of generalization bounds for the induced classifiers.

The SVMTool is intended to comply with all the requirements of modern NLP technology, by combining simplicity, flexibility, robustness, portability and efficiency with state–of–the–art accuracy. This is achieved by working in the SVM learning framework, and by offering NLP researchers a highly customizable sequential tagger generator.

2.3.1.2.2.2 Unsupervised POS Tagging

Unlike the supervised models, the unsupervised POS tagging models do not require a pre-tagged corpus. Instead, they use advanced computational methods like the Baum-Welch algorithm to automatically induce tagsets, transformation rules etc. Based on the information, they either calculate the probabilistic information needed by the stochastic taggers or induce the contextual rules needed by rule-based systems or transformation based systems.

Transformation-based POS tagging: In general, the supervised tagging approach usually requires large sized pre-annotated corpora for training, which is difficult for most of the cases. But recently, good amount of work has been done to automatically induce the transformation rules. One approach to automatic rule induction is to run an untagged text through a tagging model and get the initial output. A human then goes through the output of this first phase and corrects any erroneously tagged words by hand. This tagged text is then submitted to the tagger, which learns correction rules by comparing the two sets of data. Several iterations of this process are sometimes necessary before the tagging model can achieve considerable performance. The transformation based approach is similar to the rule based approach in the sense that it depends on a set of rules for tagging.

Transformation-Based Tagging, sometimes called Brill tagging, is an instance of the TBL approach to machine learning (Brill, 1995) and draws inspiration from both the rule-based and stochastic taggers. Like the rule-based taggers, TBL is based on rules that specify what tags should be assigned to a particular word. But like the stochastic taggers, TBL is a machine learning technique, in which rules are automatically induced from the data.

2.3.2 POS Taggersin Indian Languages: A Literature Survey

Compared to Indian languages, foreign languages like English, Arabic and other European languages have many POS taggers. POS tagging is generally classified into rule-based systems, probabilistic data-driven systems, neural network systems or hybrid systems. Many POS taggers were also developed using machine learning techniques [38] such as Support Vector Machine models, HMMs, transformation based error driven learning, decision trees, maximum entropy methods, conditional random fields. Literature shows that, for Indian languages, POS taggers were developed only in Hindi, Bengali, Panjabi and Dravidian languages. As per our knowledge, no other publicly available attempts are available in other Indian languages.

2.3.2.1 POS Taggers for Hindi Language

A number of POS taggers were developed in Hindi language using different approaches. In the year 2006, three different POS tagger systems were proposed based on Morphology driven, ME and CRF approaches respectively. There are two attempts for POS tagger developments in 2008, both are based on HMM approaches and proposed by Manish Shrivastava and Pushpak Bhattacharyya. Nidhi Mishra and Amit Mishra proposed a Part of Speech Tagging for Hindi Corpus in 2011. In an another attempt, a POS tagger algorithm for Hindi was proposed by Pradipta Ranjan Ray, Harish V., Sudeshna Sarkar and Anupam Basu.

i) In the first attempt, Smriti Singh proposed a POS tagging methodology which can be used by languages having lack of resources [39]. The POS tagger is built based on hand-crafted morphology rules and does not involve any sort of learning or disambiguation process. The system makes use of locally annotated modestly-sized corpora of 15,562 words, exhaustive morphological analysis backed by high-coverage lexicon and a decision tree based learning algorithm called CN2. The tagger also uses the affix information stored in a word and assigns a POS tag by taking in consideration the previous and the next word in the Verb Group (VG) to correctly identify the main verb and the auxiliaries. The system uses Lexicon lookup for identifying the other POS categories. The performance of the system was evaluated by a 4-fold cross validation over the corpora and found 93.45% accuracy.

ii) Aniket Dalal, Kumar Nagaraj, Uma Sawant and Sandeep Shelke, proposed a POS tagger based on ME based approach [39]. To develop a POS tagger based on ME approach requires feature functions extracted from a training corpus. Normally a feature function is a boolean function which captures some aspect of the language which is relevant to the sequence labelling task. The experiment showed that the performance of the system depend on size of the training corpus. There is an increase in performance till it reaches 75% of the training corpus after which there is a reduction in accuracy due to over fitting of the trained model to training corpus. The least and best POS tagging accuracy of the system was found to be 87.04% and 89.34% and the average accuracy over 10 runs was 88.4%.

iii) The third POS tagger is based Conditional Random Fields developed by Agarwal Himashu and Amni Anirudh in 2006 [39]. This system makes use of Hindi morph analyzer for training purpose and to get the root-word and possible POS tag for every word in the corpus. The training is performed with CRF++ and the training data also contains other information like suffixes, word length indicator and special characters. A corpus size of 1, 50,000 words were used for training and testing purposes and accuracy of the system was 82.67%.

iv) The HMM based approach was intended to utilize the morphological richness of the languages without resorting to complex and expensive analysis [39]. The core idea of this approach was to explode the input in order to increase the length of the input and to reduce the number of unique types encountered during learning. This idea increases the probability score of the correct choice and at the same time decreasing the ambiguity of the choices at each stage. Data sparsity also decreases by new morphological forms for known base words. Training and testing were performed with an exploded corpus size of 81751 tokens, which were divided into 80% and 20% parts respectively.

v) An improved Hindi POS tagger was developed by employing a naive (longest suffix matching) stemmer as a pre-processor to the HMM based tagger [40]. Apart from a list of possible suffixes, which can be easily created using existing machine learning techniques for the language, this method does not require any linguistic resources. The reported performance of the system was 93.12%.

vi) Nidhi Mishra and Amit Mishra proposed a Part of Speech Tagging for Hindi Corpus in 2011 [41]. In the proposed method, the system scans the Hindi corpus and then extracts the sentences and words from the given corpus. Also the system search the tag pattern from database and display the tag of each Hindi word like noun tag, adjective tag, number tag, verb tag etc.

vii) Based on lexical sequence constraints, a POS tagger algorithm for Hindi was proposed by Pradipta Ranjan Ray, Harish V., Sudeshna Sarkar and Anupam Basu [42]. The proposed algorithm acts as the first level of part of speech tagger, using constraint propagation, based on ontological information, morphological analysis information and lexical rules. Even though the performance of the POS tagger has not been statistically tested due to lack of lexical resources, it covers a wide range of language phenomenon and accurately captures the four major local dependencies in Hindi.

2.3.2.2 POS Taggers for Bengali

A substantial amount of work has already been done in POS tagger developments for Bengali language using different approaches. In the year 2007, two stochastic based taggers were proposed by Sandipan Dandapat, Sudeshna Sarkar and Anupam Basu using HMM and ME approaches. Also Ekbal Asif developed a POS tagger for Bengali language using CRF. In 2008, Ekbal Asif and Bandyopadhyay S developed another machine learning based POS tagger using SVM algorithm. An Unsupervised Parts-of-Speech Tagger for the Bangla language was proposed by Hammad Ali in 2010. Debasri Chakrabarti of CDAC Pune proposed a Layered Parts of Speech Tagging for Bangla in 2011.

i) In the first attempt three different types of stochastic POS taggers were developed. In this attempt a supervised and semi supervised bigram HMM & a ME based model was explored based on tagset of 40 tags [39]. The first model called as HMM-S makes use of the supervised HMM model parameters where as the second uses the semi supervised model parameters and is called HMM-SS. A manually annotated corpus of about 40,000 words was used for both supervised HMM and ME model. For testing a set of randomly selected 5000 words have been used for all three cases and the results showed that, the supervised learning model outperforms over other models. They also showed that further improvement can be achieved by incorporating a morphological analyzer for any model.

ii) The second POS tagger is based on CRF framework, where features selection plays an important role in the development of POS tagger [43, 39]. A tagset of 26 tags were used to develop the POS tagger. In this approach the system makes use of the different contextual information of the words along with the variety of features that are helpful in predicting the various POS classes. For training purpose a corpus size of 72,341 tagged words were used. The system was tested with 20,000 words selected from out of corpus and achieved 90.3%.

iii) The third POS tagger for Bengali is based on statistical approach using a supervised machine learning algorithm called SVM [38,39]. The earlier CRF based corpus was used for training and testing the POS tagging system using SVM based algorithm. The entire training corpus was divided into two different set of sizes 57,341 and 15000 words each and used as a training and development set. The test data of CRF model was used to evaluate the performance of the SVM based system and reported 86.84% accuracy.

iv) In the year 2010, Hammad Ali proposed an unsupervised POS tagger for the Bangla language based on a Baum-Welch trained HMM approach [45]. The proposed Layered Parts of Speech Tagger is a rule based system, with four levels of layered tagging [43]. The tagset used in the POS tagger was based on common tag set for Indian Languages and IIIT tagset guidelines. In the first level, a universal category containing 12 different categories are identified which is used to assign ambiguous basic category of a word. Followed by the first level, disambiguation rules are applied in the second level with detailed morphological information. The third and fourth levels are intended to tagging of multi word verbs and local word grouping. The proposed rule based approach shows better performance.

2.3.2.3 POS Taggers for Punjabi Language

There is only one publicly available attempt proposed in POS tagger for Panjabi language [39]. Using rule based approach, a Panjabi POS tagger developed by Singh Mandeep, Lehal Gurpreet, and Sharma Shiv, in 2008. The fine–grained tagset containsaround 630 tags, which consists of all the tags for the various word classes, word specific tags, and tags for punctuations. The proposed tagger uses only handwritten linguistic rules which are used to disambiguate the parts-of-speech information for a given word, based on the context information. Using the rule based disambiguation approach a database was designed to store the rules. To make the structure of verb phrase more understandable four operator categories have been established. Also a separate database is maintained for marking verbal operator. The performance of the system was manually evaluated to mark the correct and incorrect tag assignments and the system reports an accuracy of 80.29% including unknown words and 88.86% excluding unknown words.

2.3.2.4 POS Taggers for South Dravidian Languages

Some noticeable POS taggers developments were done in Dravidian languages like Tamil, Telugu, Malayalam and Kannada languages. There are six different attempts for Tamil, three for Telugu and two attempts in case of Malayalam. There is no publicly noticeable POS tagger development in case of Kannada language.

2.3.2.4.1 POS Taggers for Tamil

There are six different attempts for the development in POS tagger for Tamil language. Vasu Ranganathan proposed a Tamil POS tagger based on Lexical phonological approach. Another POS tagger was prepared by Ganesan based CIIL Corpus and tagset. An improvement over a rule based Morphological Analysis and POS Tagging in Tamil were developed by M. Selvam and A.M. Natarajan in 2009. Dhanalakshmi V, Anand Kumar, Shivapratap G, Soman KP and Rajendran S of AMRITA University, Coimbatore developed two POS taggers for Tamil using their own developed tagset in 2009.

i) Vasu Ranganathan developed a POS tagger for Tamil called ‘Tagtamil’ based on Lexical phonological approach [46]. Morphotactics of morphological processing of verbs was performed using index method. The advantages of Tagtamil POS tagger is that, it handle both tagging and generation.

ii) The second Tamil POS tagger was based on CIIl corpus and proposed by Ganesan [46]. He used his own tagset, and he tagged a portion of CIIL corpus by using a dictionary as well as a morphological analyzer. Manual correction was performed and trained the system repeatedly in order to increase the performance of the system. The tags are added morpheme by morpheme. Its efficiency in other corpora has to be tested.

iii) The third POS tagger system was proposed by Kathambam using heuristic rules based on Tamil linguistics for tagging, without using either the dictionary or the morphological analyzer [46]. The system used twelve heuristic rules and identifies the tags based on PNG, tense and case markers. Using a list of words in the tagger, the system check for standalone words. Unknown words are tagged using 'Fill in rule' by using bigram approach.

iv) Using Projection and Induction techniques, an improved rule based morphological analysis and POS Tagging in Tamil was proposed by M. Selvam and A.M. Natarajan in 2009 [47]. Rule based techniques cannot address all inflectional and derivational word forms. There for improvement of rule based morphological analysis and POS tagging through statistical methods like alignment, projection and induction is essential. The proposed idea was based on this purpose and achieved an improved accuracy of about 85.56%. Using an alignment-projection techniques and categorical information, a well organized POS tagged sentences in Tamil were obtained for the Bible corpus. Through alignment, lemmatization and induction processes, root words were induced from English to Tamil. Root words obtained from POS projection and morphological induction, further improved the accuracy of the rule based POS tagger.

v) Dhanalakshmi V, Anand Kumar, Shivapratap G, Soman KP and Rajendran S of AMRITA University, Coimbatore developed a POS tagger for Tamil using Linear Programming approach [48]. They have developed their own POS tagset consists of 32 tags and used in their POS tagger model. They have proposed a SVM methodology, based on Linear Programming for implementing automatic Tamil POS tagger. A corpus of twenty five thousand sentences is trained with linear programming based SVM. The testing was performed using 10,000 sentences and reported an overall accuracy of 95.63%.

vi) In another attempt they had developed a POS tagger using machine learning techniques, where the linguistical knowledge is automatically extracted from the annotated corpus [49]. The same tagset was used here also to develop POS tagger.

This is a corpus based POS tagger and annotated corpus size of two hundred and twenty five thousand words was used for training (1, 65,000 words)) and testing (60,000 words) the accuracy of the POS tagger. Support vector machine algorithms were used to train and test the POS tagger system and reported an accuracy of 95.64%.

2.3.2.4.2 POS Taggers for Telugu Language

NLP in Telugu language is in better position when compared with other South Dravidian and many of other Indian languages. There are three noticeable POS tagger developments in Telugu, based on Rule-based, Transformation based learning and Maximum Entropy based approaches [39]. An annotated corpus of 12000 words was constructed to train the transformation based learning and Maximum Entropy based POS tagger models. The existing Telugu POS tagger accuracy was also improved by a voting algorithm by Rama Sree, R.J. and Kusuma Kumari P in 2007.

i) The rule based approach uses various functional modules which works together to give tagged Telugu text [39]. Tokenizer, Morphological Analyzer, Morph-to-POS translator, POS disambiguator, unigram, bigram rules and Annotator are the different functional modules used in the system. The function of Tokenizer is to separate pre-edited input text into separate sentences and each sentence to words. These words are then given to MA for analysis of each word. Pattern rule based Morph-to-POS translator then converts morphological analysis into their corresponding tags. This is followed by handling the disambiguation problem by the POS disambiguator which reduces the problem of POS ambiguity. Using unigram and bigram rules ambiguity is controlled in the POS tagger system. Annotator is used to produce the tagged words in a text and reported accuracy of the system was 98%.

ii) In the second attempt, Brill Transformation Rule Based Learning (TRBL) was used to build a POS tagger for Telugu language [39]. The Telugu language POS tagger system consists of three phases of Brill tagger: Training, Verification and Testing. The reported accuracy of the proposed POS tagger is 90%.

iii) Another Telugu POS tagger was also developed based on Maximum Entropy approach [39]. The idea behind the ME approach is similar to the general principles used in other languages. The proposed POS tagger was implemented using publicly available Maximum Entropy Modelling toolkit [MxEnTk] and the reported accuracy is 81.78%.

2.3.2.4.3 POS Taggers for Malayalam

There are two separate corpus based POS tagger for Malayalam language which were proposed as follows:

i) In 2009, Manju K., Soumya S and Sumam Mary Idicula proposed a stochastic HMM based part of speech tagger. A tagged corpus of about 1,400 tokens were generated using a morphological analyzer and trained using the HMM algorithm. An HMM algorithm in turn generated a POS tagger model that can be used to assign proper grammatical category to the words in a test sentence. The performance of the developed POS Tagger is about 90% and almost 80% of the sequences generated automatically for the test case were found correct.

ii) The second POS tagger is based on machine learning approach in which training, testing and evaluations are performed with SVM algorithms developed by Antony P.J, Santhanu P Mohan and Dr. Soman K.P of AMRITA university Coimbatore in 2010 [50]. They have proposed a new AMRITA POS tagset and based on the developed tagset a corpus size of about 180,000 tagged words were used for training the system. The performance of the SVM based tagger achieves 94 % accuracy and showed an improved result than HMM based tagger.

2.3.2.5 Generic POS Taggers for Hindi, Bengali and Telugu

Many different attempts have beenmade for developing POS tagger for three different languages namely Hindi, Bengali and Telugu in Shallow Parsing Contest for South Asian Languages in 2007 [45]. All the participants in this contest were given corpus of 20000 and 5000 words respectively for training and testing based on the IIT POS tagset which consists of 24 tags. In this contest, participants proposed eight different POS tagger development techniques. Half of the these ideas are based on HMMs technique and others used Two Level Training based, Naive Bayes, Decision Trees to Maximum Entropy Model and Conditional Random Fields for developing POS tagger. Even though all the HMM based approaches used Trigrams'n'Tags or the TnT tagger for POS tagging, there was a considerable differences in the accuracies. The noticeable fact is that no one used rule based approach to develop POS tagger in their contribution. The following section gives a brief description about each and every proposed POS tagger system.

i) G.M. Ravi Sastry, Sourish Chaudhuri and P. Nagender Reddy used HMMs for developing POS tagging[52]. They used “Trigrams'n'Tags” or the TnT tagger for their proposed system. The advantage of TnT is that it is not optimized for a particular language and the system incorporates several methods of smoothing and of handling unknown words which improved the POS tagger performance. The second HMM based generic POS tagger was developed by Pattabhi and his team [53]. They used linguistic rules to tag words for which the emission or transition probabilities are low or zero instead smoothing. Another HMM based approach was proposed by Asif and team. They are also avoided smoothing and for unknown words, the emission probability was replaced by the probability of the suffix for a specific POS tag. The final HMM based generic POS tagger was developed by Rao and Yarowsky. They have developed HMM based POS tagger along with other systems using different approaches. They used TnT based HMM, compared the result with other systems and found that HMM based system outperforms.

ii) Naive Bayes Classifier, A suffix based Naive Bayes Classifier and QTag are the other three approaches used by Rao and Yarowsky to develop generic POS tagger. A suffix based Naive Bayes Classifier uses a suffix dictionary information for handling unseen words.

iii) For modelling the POS tagger, Sandipan and team used Maximum Entropy approach and result shows that this approach is best suited to Bengali language. They used contextual features covering a word window of one and suffix and prefix information with lengths less than or equal to four. The output of the tagger for a word is restricted by using a manually built morphological analyser.

iv) In another attempt, Himanshu and his team used a CRF based approach to develop the POS taggers. In their system, they used a feature set including a word window of tow, suffixes information with length less than or equal to four, word length and flag indicating special symbols. A knowledge database was used to handle data sparsity by picking word & tag pairs which are tagged with high confidence by the initial model over a raw corpus of 150,000 words. Similar to the ME proposed by Sandipan, a set of tags listed in the knowledge database and the training data are used to restrict the output of the tagger for each word instead. The experiment results shows that the CRF approach is well suited for Bengli and Telugu and not performed well for Hindi.

v) A two level training approach based POS tagger model was proposed by Avinesh and Karthik. In this approach a TBL was applied on top of a CRF based model. Morphological information like root word, all possible categories, suffixes and prefixes are used in the CRF model along with exhaustive contextual information with a window size of 6, 6 and 4 for Hindi, Bengali and Telugu respectively. The system performance is good for Hindi and Telugu when compare with Bengali.

vi) Using a Decision Forests approach, Satish and Kishore proposed POS tagging with some innovative features based on sub words (syllables, phonemes and Onset-Vocal-Code for syllables) for Indian languages like Hindi, Bengali and Telugu. The sub words are an important source of information to determine the category of the word in Indian language and the performance of the system is encouraging only for Telugu.

2.3.3 Development of POS Tagset for Indian Languages

A number POS tagsets are developed by different organization and persons based on the general principles of tagset design strategy. However, most of the tagsets are language specific and some of these tagset are constructed by considering general peculiarities of Indian languages. A few tagset attempts were based on the feature of South Dravidian languages and other aim to a particular language. The following section gives a brief description of tagsets developed for Indian languages.

i) In a major work, IIT Hyderabad developed a Tagset in 2007, after consultations with several institutions through two workshops [51]. The aim was to create a general standard tagset suitable for all the Indian languages. The tagset also consist of a detail description of the various tags used and elaborates the motivations behind the selection of these tags. The total number of tags in the tagset is 25.

ii) The 6th Workshop on Asian Language Resources, 2008 was intended to design a common POS-Tagset framework for Indian Languages [54,55]. It was a shared work from experts from various organizations like Microsoft Research-India, Delhi

University, IIT- Bombay, Jawaharlal Nehru University- Delhi, Tamil University-Thanjavur and AU-KBC Research Centre, Chennai. There are three levels of tagsets were proposed and the top level consists 12 universal categories for all Indian languages and hence these are obligatory for any tagsets. The other levels consist of tags which are recommended and optional categories for verbs and participles.

iii) Dr.Rama Sree R.J, Dr.Uma Maheswara Rao G and Dr. Madhu Murthy K.V proposed a Telugu tagset by carefully analyzing the two tagsets developed by IIIT, Hyderabad and CALTS, Hyderabad in 2008[54]. The proposed tagset was developed based on the argument that an inflectional language needs additional tags. They proposed some additional tags over the existing tagset to capture and provide finer discrimination of the semantic content of some of the linguistic expressions.

iv) Dhanalakshmi V, Anand Kumar, Shivapratap G, Soman KP and Rajendran S of AMRITA university, Coimbatore developed a tagset for Tamil in 2009, called AMRITA tagset which consists of 32 tags [48].

v) Vijayalaxmi .F. Patil developed a POS tagset for Kannada language in 2010 which consists 39 tags [44]. This tagset was developed by considering the morphological as well as syntactic and semantic features of the Kannada language.

vi) Antony P J, Santhanu P Mohan and Soman KP of AMRITA University, Coimbatore developed a tagset for Malayalam language in 2010. The developed tagset is based on AMRITA tagset which consists of 29 tags [51].

vii) Central Institute of Indian Language (CIIL) proposed a tagset for Hindi language based on Penn tagset [55]. This tagset was designed to include more lexical categories than IIIT-Hyderabad and containing 36 tags.

viii) IIT- Karagpur developed a tagset for Bengali language which consists of 40 tags [56]. Another tagset called CRBLP tagset which consists of a total of 51 tags, where 42 tags are general POS tags, and 9 other tags are intended for special symbols.

2.4 MORPHOLOGICAL APPROACHES AND SURVEY FOR INDIAN LANGUAGES

Morphology plays an essential role in MT and many other NLP applications. Developing a well fledged MAG tools for highly agglutinative languages is a challenging task. The function of morphological analyzer is to return all the morphemes and their grammatical categories associated with a particular word form. For a given root word and grammatical information, morphological generator will generate the particular word form of that word.

The morphological structure of an agglutinative language is unique and capturing its complexity in a machine analyzable and generatable form is a challenging job. Analyzing the internal structure of a particular word is an important intermediate stage in many NLP applications especially in bilingual and multilingual MT system. The role of morphology is very significant in the field of NLP, as seen in applications like MT, QA system, IE, IR, spell checker, lexicography etc. So from a serious computational perspective the creation and availability of a morphological analyzer for a language is important. To build a MAG for a language one has to take care of the morphological peculiarities of that language, specifically in case of MT. Some peculiarities of language such as, the usage of classifiers, excessive presence of vowel harmony etc. make it morphologically complex and thus, a challenge in NLG.

The literature shows that development of morphological analysis and generation work has been successfully done for languages like English, Chinese, Arabic and European languages using various approaches from last few years. A few attempts have been made in developing morphological analysis and generation tool for Indian languages. Even though the morphological analyzer and generator play an important role in MT, literature shows that it is still an ongoing process for various Indian languages. This paper addresses the various approaches and developments in morphological analyzer and generator for Indian language.

The first sub section of this section discusses various approaches that are used for building morphological analyzer and generator tool for Indian languages. The second sub section gives a brief explanation about different morphological analyzer and generator developments for Indian languages.

2.4.1 Morphological Analyzer and Generator Approaches

There are many language dependent and independent approaches used for developing morphological analyzer and generator [63]. These approaches can also be classified generally into corpus based, rule based and algorithmic based. The corpus based approach, where a large sized well generated corpus is used for training with a machine learning algorithm. The performance of the system depends on the feature and size of the corpus. The disadvantage is that corpus creation is a time consuming process. On the other hand, rule based approaches are based on a set of rules and dictionary that contains root and morphemes. In rule based approaches every rule depends on the previous rule. So if one rule fails, it will affect the entire rules that follow it. When a word is given as an input to the morphological analyzer and if the corresponding morphemes are missing in the dictionary then the rule based system fails. Literature shows that there are number of successful morphological analyzer and generator development for languages like English, Chinese, Arabic and European languages using these approaches [24]. Recent developments in Indian language NLP shows that many morphological analyzer and generators are created successfully using these approaches. Brief descriptions of most commonly used approaches are as follows:

2.4.1.1 Corpus Based Approach

In case of corpus based approach, a large sized well generated corpus is required for training. Any machine learning algorithm is used to train the corpus and collect the statistical information and other necessary features from the corpus. The collected information is used as a MAG model. The performance of the system will depend on the feature and size of the corpus. The disadvantage is that corpus creation is a time consuming process. This approach is suitable for languages having well organized corpus.

2.4.1.2 Paradigm Based Approach

For a particular language, each word category like nouns, verbs, adjectives, adverbs and postpositions will be classified into certain types of paradigms. Based on their morphophonemic behavior, a paradigm based morphological compiler program is used to develop MAG model. In the paradigm approacha linguist or the language expert is asked to provide different tables of word forms covering the words in a language. Based on this information and the feature structure with every word form, a MAG can be built. The paradigm based approach is also well suited for highly agglutinative language nature. So paradigm based approach or the variant of this scheme has been used widely in NLP. Literature shows that morphological analyzers are developed for almost all Indian languages using paradigm based approach.

2.4.1.3 Finite State Automata Based Approach

Finite state machine or Finite State Automation (FSA) or finite automation uses regular expressions to accept or reject a string in a given language [64]. In general, an FSA is used to study the behavior of a system composing of state, transitions and actions. When FSA starts working, it will be in the initial stage and if the automation is in any one of the final state it acceptsthe input and stops working.

2.4.1.4 Two- Level Morphology Based Approach

In 1983, Kimmo Koskenniemi, a Finnish computer scientist developed a general computational model for word-form recognition and generation called Two- level morphology [64]. This development was one of the major breakthroughs in the field of morphological parsing, which is based on morphotactics and morphophonemics concepts. The advantage of two- level morphology is that the model does not depend on a rule compiler, composition or any other finite-state algorithm. The "two-level" morphological approach consists of two levels called lexical and surface form and a word is represented as a direct, letter-for-letter correspondence between these forms. The Two-level morphology approach is based on the following three ideas:

- Rules are symbol-to-symbol constraints that are applied in parallel, not sequentially like rewrite rules.
- The constraints can refer to the lexical context, to the surface context, or to both contexts at the same time.
- Lexical lookup and morphological analysis are performed in tandem.

2.4.1.5 FST Based Approach

FST is a modified version of FSA by accepting the principles of a two level morphology. A FST, essentially is a finite state automaton that works on two (or more) tapes. The most common way to think about transducers is as a kind of “translating machine” which works by reading from one tape and writing onto the other. FST’s can be used for both analysis and generation (they are bidirectional) and it acts as a two level morphology. By combining the lexicon, orthographic rules and spelling variations in the FST, we can build a morphological analyzer and generator at once.

2.4.1.6 Stemmer Based Approach

Stemmer uses a set of rules containing list of stems and replacement rules to stripping of affixes. It is a program oriented approach where the developer has to specify all possible affixes with replacement rules. Potter algorithm is one of the most widely used stemmer algorithm and it is freely available. The advantage of stemmer algorithm is that it is very suitable to highly agglutinative languages like Dravidian languages for creating MAG.

2.4.1.7 Suffix Stripping Based Approach

Highly agglutinative languages such as Dravidian languages, a MAG can be successfully built using suffix stripping approach. The advantage of the Dravidian language is that no prefixes and circumfixes exist for words. Words are usually formed by adding suffixes to the root word serially. This property can be well suited for suffix stripping based MAG. Once the suffix is identified, the stem of the whole word can be obtained by removing that suffix and applying proper orthographic (sandhi) rules. A set of dictionaries like stem dictionary, suffix dictionary and also using morphotactics and sandhi rules, a suffix stripping algorithm successfully implements MAG.

2.4.1.8 Directed Acrylic Word Graph Based Approach

Directed Acrylic Word Graph (DAWG) is a very efficient data structure that can be used for developing both morphological analyzer and generator. DAWG is language independent and it does not utilize any morphological rules or any other special linguistic information. But DAWG is very suitable for lexicon representation and fast string matching, with a great variety of application. Using this approach, the University of Partas Greece developed MAG for Greek language for the first time. There after the method was applied for other languages including Indian languages.

2.4.2 MAG Developments in Indian Languages: A Literature Survey

In general there are several attempts for developing morphological analyzer and generator all over the world using different approaches. In 1983 Kimmo Koskenniemi developed a two-level morphology approach, where he tested this formalism for Finnish language [57]. In this two level representation, the surface level is to describe the word forms as they occur in written text and the lexical level is to encode lexical units such as stem and suffixes. In 1984 the same formalism was extended in other languages such as Arabic, Dutch, English, French, German, Italian, Japanese, Portuguese, Swedish, Turkish and developed morphological analyzers successfully. In the same time a rule based heuristic analyzer for Finnish nominal and verb forms was developed by Jappinen [58]. In 1996, Beesley [59] developed an Arabic FST for MA using Xerox FST (XFST), by reworking extensively on the lexicon and rules in the Kimmo-style. At 2000, Agirve introduced a word–grammar based morphological analyzer using the two- level and a unification- based formalism for a highly agglutinative language called Basque [60]. Similarly using XFST, Karine made a Persian MA [61] in 2004 and Wintner came up with a morphological analyzer for Hebrew [62] in 2005. Oflazer Kamel developed a FSM based Turkish morphological analyzer. In 2008, using the syllables and utilizing the surface level clues, the features present in a word are identified for Swahili (or Kiswahili) language by Robert Elwell.

Extensive work has been done already for developing MAG in various Indian languages from last ten to fifteen years. Even though there have been many attempts in developing MAG for Indian languages, only few works are publicly focused. During the literature survey, following different attempts were found for developing MAG for Indian language NLP.

2.4.2.1 MAG for Tamil Language

Literature shows that the majority of the work in Indian language morphological analyser and generator was done in Tamil language. The following are the noticeable developments in Tamil MAG. The first five MAG are the recent developments where as the remaining MAG were developed before the year 2007.

i) AMRITA Morph Analyzer and generator for Tamil- A Rule Based Approach (2010): Dr. A.G. Menon, S. Saravanan, R. Loganathan and Dr. K. Soman, Amrita University, Coimbatore, developed a rule based Morphological Analyzer and generator for Tamil using FST called AMAG [65]. The performance of the system is based on lexicon and orthographic rules from a two level morphological system. The system consists of a list of 50000 nouns, around 3000 verbs and a relatively smaller list of adjectives. The proposed AMAG is compared with the existing Tamil morph analyzer and generator called ATCHARAM and has exhibited better performance.

ii) A Novel Algorithm for Tamil Morphological Generator (2010): M.Anand Kumar, V.Dhanalakshmi and Dr. K P Soman, CEN, Amrita University, Coimbatore developed a morphological generator for Tamil based on suffix stripping algorithm [66]. The system consists of two modules, in which the first module handles the lemma/root part and the second module handles the Morpho-lexical information. The system requires the following information: morpho-lexical information file, suffix table, paradigm classification rules and stemming rules. Based on a simple, efficient and language independent algorithm and with less amount data, the system efficiently handles compound words, transitive and intransitive verbs and also the proper nouns.

iii) An Improvised Morphological Analyzer cum Generator for Tamil (2010): This work is proposed by Parameswari K, CALTS, university of Hyderabad and deals with the improvised database implemented on Apertium for morphological analysis and generation [67,68]. The improvised MAG uses the FSTs algorithm for one-pass analysis and generation, and the Word and Paradigm based database. The system performance is measured and compared for its speed and accuracy with the other available Tamil Morphological analyzers which were developed in CALTS, Hyderabad and AU-KBC research Centre, Anna University. Experiment result showed that the proposed MAG performs better than the other.

iv) A Sequence Labelling Approach to Morphological Analyzer for Tamil (2010): Anand Kumar M, Dhanalakshmi V, Soman K.P and Rajendran S of AMRITA Vishwa Vidyapeetham, Coimbatore, developed a morphological analyzer for Tamil language based sequence labelling approach [69]. In the proposed work morphological analyzer problem is redefined as classification problem and solved using machine learning methodology. This is a corpus based approach, where training and testing is performed with support vector machine algorithms. The training corpus consists of 130,000 verb words and 70,000 noun words respectively. The system is tested with 40000 verbs and 30000 nouns taken from Amrita POS Tagged corpus. The performance of the system was also compared with other systems developed using the same corpus and results showed that SVM based approach outperforms the other.

v) FSA-based morphological generator for Tamil (2010): A finite state automata based morphological generator is developed by Menaka S, Vijay Sundar Ram and Sobha Lalitha Devi [70]. Two separate experiments were conducted to evaluate the system for nouns and verbs using both correct and wrong inputs. The experiment showed that finite-state based morphological generator is well-suited for highly agglutinative and inflectional languages like Tamil.

vi) Rajendran's Morphological Analyzer for Tamil: The first step towards a preparation of morphological analyzer for Tamil was initiated by 'Anusaraka'group of researchers under the guidance of Dr Rajendran [46], Tamil University, Tanjavoor. 'Anusaraka' is a MT project intended for translation between Indian languages. The developed morphological analyzer for Tamil was used for Translating Tamil language into Hindi at the word level.

vii) Genesan's Morphological Analyzer for Tamil: Ganesan developed a morphological analyzer for Tamil to analyze CIIL corpus. He exploits phonological and morphophonemic rules as well as morphotactic constraints of Tamil in building morphological analyzer. Recently he has built an improved and efficient morphological parser.

viii) Kapilan's Morphological Analyzer for Tamil Verbal Forms:Anotherattempt was made by Kapilan and he prepared a morphological analyzer for verbal forms in Tamil.

ix) Deivasundaram's Morphological parser: Deivasundarm has prepared a morphological analyzer for Tamil for his Tamil Word Processor. He too makes use of phonological and morphophonemic rules and morphotnatic constraints for developing his parser.

x) AUKBC Morphological Parser for Tamil: AUKBC NLP team under the supervision of Dr Rajendran developed a Morphological parser for Tamil. The API Processor of AUKBC makes use of the finite state machinery like PCKimmo. It parses, but does not generate.

xi) Vishnavi's Morphological Generator for Tamil: Vaishnavi researched for her M.Phil dissertation on morphological generator for Tamil. The Vaishanvi's morphological generator implements the item and process model of linguistic description. The generator works by the synthesis method of PCKimmo.

xii) Ramasamy's Morphological Generator for Tamil:Ramasamy has prepared a morphological generator for Tamil for his MPhil dissertation.

xiii) Winston Cruz's Parsing and Generation of Tamil Verbs: Winston Cruz makes use of GSmorph method for parsing Tamil verbs. GSmorph too does morphotactics by indexing. The algorithm simply looks up two files to see if the indices match or not. The processor generates as many forms as it parses and uses only two files.

xiv) Vishnavi's Morphological Analyzer for Tamil:Vaishnavi again researched for her Ph.D. dissertation on the preparation of Morphological Analyzer for Tamil. She proposes a hybrid model for Tamil. It finds its theoretical basis in a blend of IA and IP models of morphology. It constitutes an in-built lexicon and involves a decomposition of words in terms of morphemes within the model to realize surface well-formed word-forms. The functioning can be described as defining a transformation depending on the morphemic nature of the word stem. The analysis involves a scanning of the string from the right to left periphery scanning each suffix at a time stripping it, and reconstructing the rest of the word with the aid of phonological and morphophonemicrules exemplified in each instance. This goes on till the string is exhausted. For the sake of comparison she implements AMPLE and KIMMO models. She concludes that Hybrid model is more efficient than the rest of the models.

xv) Dhurai Pandi's Morphological Generator and Parsing Engine for Tamil Verb Forms:It is a full-fledged morphological generator and a parsing engine on verb patterns in modern Tamil.

xvi) RCILTS-T’s Morphological analyzer for Tamil: Resource Centre for Indian Language Technological Solutions-Tamil has prepared a morphological analyzer for Tamil. It is named as ‘Atcharam’. ‘Atcharam’ takes a derived word as input and separates into root word and associated morphemes. It uses a dictionary of 20000 root words based on fifteen categories. It has two modules - noun and verb analyzer based on 125 rules. It uses heuristic rules to deal with ambiguities. It can handle verb and noun inflections.

xvii) RCILTS-T’s Morphological generator for Tamil: Resource Centre for Indian Language Technological Solutions-Tamil also developed a morphological generator for Tamil. It is named as ‘Atchayam’. ‘Atchayam’ generates words when Tamil morphs are given as input. It has two major modules – noun and verb generators. The noun section handles suffixes like plural markers, oblique form, case markers and postpositions. The verb section takes tense and PNG makers, relative and verbal participle suffixes, and auxiliary verbs. It uses sandhi rules and125 morphological rules. It handles adjectives and adverbs. It has word and sentence generator interfaces.

2.4.2.2 MAG for Kannada Language

There are three different developments in Kannada language MAG. The first attempt was made by T. N. Vikram and Shalini R Urs in the year 2007 and they developed a morphological analyzer prototype model based on finite state machine. The R V College of Engineering, Bangalore proposed another morphological analyzer and generator using Trie data structure. Using Network and Process Model, University of Hyderabad developed a MAG system for Kannada language.

i) T. N. Vikram and Shalini R Urs developed a prototype of morphological analyzer for Kannada language (2007) based on Finite State Machine [3]. This is just a prototype and does not handle compound formation morphology and can handle maximum 500 distinct nouns and verbs.

ii) Kannada Morphological Analyzer and Generator Using Trie (2011): Using rule based with paradigm approach, Shambhavi. B. R and Dr. Ramakanth Kumar P of R V College of Engineering, Bangalore proposed a morphological analyzer and generator for Kannada language [24]. They used Trie as a data structure for the storage of suffixes and root words. The disadvantage of Trie is that it consumes more memory as each node can have at most ‘y’ children, where ‘y’ is the alphabet count of the language. As a result it can handle up to maximum 3700 root words and around 88K inflected words.

iii) MORPH- A network and process model for Kannada morphological analysis/ generation was developed by K. Narayana Murthy and the reported performance of the system is 60 to 70% on general texts[24]. The advantage of finite state network is that, it captures all the affixes, their ordering and the various combinations permitted in a declarative and bidirectional fashion. Since the same network is used both for analysis and generation, it reduces the overall overheads of the system.

2.4.2.3 MAG for Malayalam Language

In case of Malayalam, there are two different developments in MAG as follows:

i) Malayalam Morphological Analyser and Tamil Morphological Generator for Malayalam - Tamil MT (2011): Based on suffix stripping and suffix joining approach, using a bilingual dictionary, a Malayalam morphological analyzer and a Tamil morphological generator have been developed by Jisha P.Jayan, Rajeev R and Dr. S Rajendran [71]. The developed analyzer and generator are used for Malayalam - Tamil MT.

ii) Morphological analyzer for Malayalam verbs (2008): Saranya S.K and Dr Soman K P of AMRITA Vishwa Vidyapeetham, Coimbatore developed a prototype morphological analyzer for Malayalam language based on hybrid approach of Paradigm and Suffix Stripping Method [64].

2.4.2.4 MAG for Hindi Language

Even though there are many attempts in MAG for Hindi language, only one development is available publicly. Teena Bajaj proposed a method for extending the range of existing morphological analyzer system for Hindi language [72]. The work focuses on how the strength ofthe existing morph analyzer can be improved by merging it with a semi-supervised approach for learning of Hindi morphology.

2.4.2.5 MAG for Punjabi Language

There are two different attempts to develop Morphological Analyzer and Generator for Panjabi language [73]. Under 'Anusarka' project, IIIT Hyderabad developed a Punjabi Morph at first time. Later Dr Mandeep Singh, Advanced Centre for Technical Development of Punjabi Language, Punjabi University developed a MAG for Panjabi language.

2.4.2.6 MAG for Bengali Language

Development of a morphological analyser for Bengali (2009): An open-source morphological analyser for Bengali Language using finite state technology was developed by Abu Zaher Md. Faridee and Francis M. Tyers [74]. This is the first open source attempt in creating a fully-functional morphological analyser and the system is currently under development stage.

2.4.2.7 MAG for Assamese, Bengali, Bodo and Oriya Languages

Morphological analyzer using rule based affix stripping approach (2011): The design and development of morphological analyzers for four Indian languages- Assamese, Bengali, Bodo and Oriya was proposed by Mona Prakash and Rajesha N, CIIL Mysore [75]. At present it is an ongoing workon dictionary based and suffix stripping approach and the performance of the system directly related to the size of the dictionary. The developed prototype model currently can handle inflectional suffixes and work progress to handle derivation as well as prefixation.

2.5 PARSING APPROACHES AND SURVEY FOR INDIAN LANGUAGES

Syntactic parsing is the task of recognizing a sentence and assigning a syntactic structure to it. A syntactic parser is an essential tool used for various NLP applications and natural language understanding. Literature survey on natural language parsing reveals that majority of the parsing techniques were developed specifically for English, Arabic and European languages using different approaches. Literature shows that the rule based grammar refinement process is extremely time consuming and difficult and also failed to analyze accurately a large corpus of unrestricted text. Hence, most modern parsers are based on statistical or at least partly statistical, which allows the system to gather information about the frequency with which various constructions occur in specific contexts. Every statistical approach requires the availability of aligned corpora which are: large, of good-quality and representative.

Parsing of sentences is considered to be an important intermediate stage for semantic analysis in NLP application such as IR, IE and QA. The study of structure of sentence is called syntax. It attempts to describe the grammatical order in a particular language in term of rules which in detail explain the underlying structure and a transformational process. Syntax provides rules to put together words to form components of sentences and to put together these components to form meaningful sentences. In NLP, syntactic parsing or more formally syntactic analysis is the process of analyzing and determining the structure of a text which is made up of sequence of tokens with respect to a given formal grammar. Because of the substantial ambiguity present in the human language, whose usage is to convey different semantics, it is much difficult to design the features for NLP tasks. The main challenge is the inherent complexity of linguistic phenomena that makes it difficult to represent the effective features for the target learning models.

Indian languages are highly inflectional, relatively free word order and agglutinative. Because of these features most of these techniques cannot be applied to Indian language context directly. Most of the Indian language parsers were developed recently and still it is an ongoing process. This section addresses the various approaches and parsing developments for Indian languages.

2.5.1 Parsing Approaches

A well known parsing approach known as Nivre’s parser was successfully implemented in a variety of languages like relatively free-word order language like Turkish, inflectionally rich language like Hindi, fixed word order language like English, and relatively case-less and less inflectional language like Swedish. Another simple approach called CFG formalism was used in languages like Dutch, Turkish and English to develop parsers. In order to suit the context of Indian languages, a formalism called ‘Paninian Grammar Framework’ was developed. Collin’s and Mc-Donald’s parser are the other well known parsing techniques. Generally, natural language parsing can be broadly classified into three categories: (i) rule based (ii) statistical based and (iii) generalized parsers [77]. All the developed parsers belong to any one of these categories and follow either ‘top-down’ or ‘bottom-up’ direction. Statistical and rule based parsing techniques are called ‘data-driven’ and ‘grammar-driven’ approaches respectively.

2.5.1.1 Rule Based Parsers

A rule- based parser uses the hard – coded rules to identify the best parse tree for a given grammar. In a rule based parsing, production rules are applied recursively and as a result overlapping problem may arise. Dynamic programming (DP) technique can be used to solve the overlapping problem efficiently. The cache for sub parse trees in the DP-based parsers is called the ‘chart’ and consequently the DP-based parsers are called ‘chart parsers’. The CYK algorithm developed by Cocke (1970), Kasami (1965), Younger (1967) and Early algorithm developed by Jurafsky and Martin, in 2000 belong to rule based parsers.

2.5.1.2 Statistical Based Parsers

The main effort in parsing of a sentence is to resolve the ambiguities. It is very hard to write complex rules to resolve such ambiguities. In contrast to the rule based approach, statistical parsing algorithms collect statistical data from correctly parsed sentences, and resolves ambiguity by experience. The advantage of statistical approach is that it covers the whole grammar usage of the language. The performance of the statistical parsers depends on training corpus used to gather statistical information about the grammar of the language. Instead of using rules to find the correct parse tree, statistical parsers select the best parse tree from possible candidates based on the statistical information. Sampson proposed the first statistical based parsing in 1986, using a manually designed language model based on a set of transition networks, and a stimulated annealing decoding search PCFG algorithm. CFG and based parsers are the examples for statistical parsers.

2.5.1.3 Generalized parsing

The framework behind both rule based and statistical parsing are similar. Using this advantage, Goodman proposed a general parsing algorithm based on semiring idea. In the year 2005, Melamed suggested another generalized parsing algorithm which was based on semiring parsing. Melamed generalized algorithm consists of five components such as:

grammar, logic, semiring, search strategy and *termination condition*. As the name suggests, grammar defines terminal and non-terminal symbols, as well as a set of production rules. Logic defines the mechanism of how the parser runs by generating new partial parse trees. The semiring defines how partial parse trees are scored. The search strategy defines the order in which partial parse trees are processed and the termination condition defines when to stop the logic necessarily.

2.5.2 Parser Developments in Indian Languages: A Literature Survey

A series of statistically based parsers for English are developed by various researchers namely: Charniak-1997, Collins-2003, Bod et al. - 2003 and Charniak and Johnson- 2005 [86, 87]. All these parsers are trained and tested on the standard benchmark corpora called Wall Street Journal (WSJ). A probability model for a lexicalized PCFG was developed by Charniak in1997. In the same time Collins described three generative parsing models, where each model is a refinement on the previous one, and achieved improved performance. In 1999 Charniak introduced a much better parser called maximum-entropy parsing approach. This parsing model is based on a probabilistic generative model and uses a maximum-entropy inspired technique for conditioning and smoothing purposes. In the same period Collins also presented a statistical parser for Czech using the Prague Dependency Treebank. The first statistical parsing model based on a Chinese Treebank was developed in 2000 by Bikel and Chiang. A probabilistic Treebank based parser for German was developed by Dubey and Keller in 2003 using a syntactically annotated corpus called ‘Negra’. The latest addition to the list of available Treebank is the ‘French Le Monde’corpus and it was made available for research purposes in May 2004. Ayesha Binte Mosaddeque & Nafid Haque wrote CFG for 12 Bangla sentences that were taken from a newspaper [76]. They used a recursive descent parser for parsing the CFG.

Even though natural language parsers play an important role in MT and other natural language applications, it is still an ongoing process for Indian languages. Comparing with foreign languages, a very little work has been done in the area of NLP for Indian languages. This section will give a brief description about various developments contributed towards natural language parsing in Indian languages.

i) In the year 2009, B.M. Sagar, Shobha G and Ramakanth Kumar P proposed a way of producing context free grammar for the Noun Phrase and Verb Phrase agreement in

Kannada Sentences [21]. In this approach, a recursive descent parser is used to parse the CFG. The system works in two stages: First, it generates the context free grammar of the sentence. In the second stage, a recursive descent parser called Recursive Descent Parser of Natural Language Tool Kit (NLTK) was used to test the grammar. As a summary, it is a grammar checking system such that for a given sentence parser says whether the sentence is syntactically correct or wrong depending upon the Noun and Verb agreement. They have tested the system using around 200 sample sentences and obtained encouraging results.

ii) Natural Language constructs for Venpa class of Tamil Poetry using Context Free Grammar was implemented by Bala Sundara Raman L, Ishwar S, and Sanjeeth Kumar Ravindranath in 2003 [79]. They used Push DownAutomata parser to parse the CFG in the proposed system.

iii) A rule based grammar checking mechanism for Hindi sentences was developed by Singh and D.K. Lobiyal in 1993 [80]. The system is suitable for all types of sentences with compound, conjunct or complex verb phrases. In the proposed model, verb and suffixes have been divided into finer subcategories to simplify the process of associating semantics with syntax. Using Lexical Functional Grammar formalism, the base grammar rules are being augmented with functional equations. This technique is used to bridge the gap between syntax and semantics of a sentence. They have developed a parser for the grammar. The grammar rules are assigned priority in a manner that the most frequently applicable rule gets higher priority than the less frequently applicable rule. The system works in such a way that, the grammar rules get fired on priority basis to make the parsing efficient.

iv) Selvam M, Natarajan A M, and Thangarajan R proposed a statistical parsing of Tamil sentences using phrase structure hybrid language model in the year 2008 [81]. They have built a statistical language model based on Trigram for Tamil language with medium of 5000 words. In the experiment they showed that statistical parsing gives better performance through tri-gram probabilities and large vocabulary size. In order to overcome some disadvantages like focus on semantics rather than syntax, lack of support in free ordering of words and long term relationship of the system, a structural component is to be incorporated. The developed hybrid language model is based on a part of speech tagset for Tamil language with more than 500 tags. The developed phrase structured Treebank was based on 326 Tamil sentences which covers more than 5000 words. The phrase structured Treebank was trained using immediate head parsing technique. Two test cases with 120 and 40 sentences have been selected from trained set and test set respectively. They reported that, the performance of the system is better than the grammar model.

v) Akshar Bharati and Rajeev Sangal described a grammar formalism called the ‘Paninian Grammar Framework’ that has been successfully applied to all free word Indian languages [82]. They have described a constraint based parser for the framework. Paninian framework uses the notion of karaka relations between verbs and nouns in a sentence. They showed that the Paninian framework applied to modern Indian languages will give an elegant account of the relation between vibhakti and karaka roles and that the mapping is elegant and compact.

vi) B.M. Sagar, Shobha G and Ramakanth Kumar P proposed a CFG Analysis for simple Kannada sentences in 2010 [21]. They have explained the writing of CFG for a simple Kannada sentence with two sets of examples. In the proposed system, a grammar is parsed with Top-Down and Bottom-Up parsers and they found that a Top-Down parser is more suitable to parse the given grammatical production.

vii) A dependency parser system for Bengali language was developed by Aniruddha Ghosh, Pinaki Bhaskar, Amitava Das and Sivaji Bandyopadhyay in 2009 [83]. They have performed two separate runs for Bengali. A statistical CRF based model followed by a rule-based post-processing technique has been used. They have used ICON 2009 datasets for training the system. They have trained the probabilistic sequence model with the morphological features like root word, POS-tag, chunk tag, vibhakti and dependency relation from the training set data. The output of the baseline CRF based system is filtered by a rule-based post-processing module by using the output obtained through the rule based dependency parser. The system demonstrated an UnlabelledAttachment Score (UAS) of 74.09%, labelled attachment score (LAS) of 53.90% and labelled accuracy score (LS) of 61.71% respectively.

viii) Sengupta,Pand B.Chaudhuri proposed a delayed syntactic-encoding-based LFG parsing strategy for an Indian Language- Bangla, in 1997 [84]. This was just an attempt in parsing of Bengali language based on the detection and formation of the proper rule set to identify characteristics of inter-chunk relations. They have tested the system on a sample of about 250 simple and complex sentences picked from newspaper clippings. Results show that, even though phrasal orderings were quite random, almost all simple sentences in active voice were correctly parsed.

ix) Parsing of Indian Languages using the freely available Malt- Parser system was developed by Joakim Nivre in 2009 [85]. He developed transition-based dependency parsers using Malt- Parser system for three Indian languages like Bangla, Hindi and Telugu. With the Malt- Parsing technique he showed that, parsing can be performed in linear time for projective dependency trees and quadratic time for arbitrary trees. A small test set of 150 sentences was used to analyse the performance of the system. The performance of the system was slightly better for Bangla and Hindi languages but for Telugu it was lower than the baseline results.

x) Hiring world's leading dependency parsers to plant Indian trees, a voting parser was proposed by Daniel Zeman in 2009 [86] called Maximum Spanning Malt (MST). The system consists of three existing, freely available dependency parsers, two of which (MST and Malt) have been known to produce state-of-the-art structures on data sets for other languages. Various settings of the parsers were explored in order to adjust them for the three Indian languages like Hindi, Bengali and Telugu, and a voting approach was used to combine them into a super parser. He showed that 'case' and 'vibhakti' are important features for parsing Hindi while their usability in Bangla and Telugu is limited by data sparseness. Based on these features, he developed best combined parsers for these languages.

xi) A constraint based Dependency parsing has been attempted and applied to a free-word order language Bangla by Sankar, Arnab Dhar and Utpal Garain in 2009 [87]. They have used a structure simplification and demand satisfaction approach to dependency parsing in Bangla language. A well known and very effective grammar formalism for free word order language called Paninian Grammatical model was used for this purpose. The main idea behind this approach was to simplify complex and compound sentential structures first, then to parse the simple structures so obtained by satisfying the 'Karaka'demands of the VGs and to rejoin such parsed structures with appropriate links and Karaka labels. A Treebank of 1000 annotated sentences was used for training the system. The performance of the system was evaluated with 150 sentences and achieves accuracies of 90.32%, 79.81%, and 81.27% for unlabelled attachments, labelled attachments and label scores, respectively.

xii) Bharat Ram Ambati, Phani Gadde and Karan Jindal explored two data-driven parsers called Malt and MST on three Indian languages namely Hindi, Telugu and Bangla in 2009 [88]. In their experiment, they merged both the training and development data and did 5-fold cross-validation for tuning the parsers. They also extracted best settings from the cross validation experiments and these settings are applied on the test data of the contest. Finally they evaluated the individual and average results on both coarse-grained and fine-grained tagset for all the three languages. They observed that for all the languages Malt performed better over MST+maxent. They also modified the implementation of MST to handle vibhakti and TAM markers for labelling. They reported that, the average of best unlabelled attachment, labelled attachment and labelled accuracies are 88.43%, 71.71% and 73.81% respectively.

xiii) A hybrid approach for parsing Bengali sentences was proposed by Sanjay Chatterji, Praveen Sonare, Sudeshna Sarkar and Devshri Roy in 2009 [89]. The system was based on data driven dependency parser. In order to improve the performance of the system, some handcrafted rules are identified based on the error patterns on the output of the baseline system.

xiv) A constraint based Hindi dependency parser was developed by Meher Vijay Yeleti and Kalyan Deepak in 2009 [90]. In the proposed system a grammar driven approach was complemented by a controlled statistical strategy to achieve high performance and robustness. The proposed system uses two stage constraint based hybrid approach to dependency parsing. They defined two stages and this division leads to selective identification and resolution of specific dependency relations at the two stages. They also used hard constraints and soft constraints to build an efficient and robust hybrid parser. From the experiment they found out that the best labelled and unlabelled attachment accuracies for Hindi are 62.20% and 85.55% respectively.

xv) Prashanth Mannem proposed a bidirectional dependency parser for Hindi, Telugu and Bangla languages in 2009 [91]. The developed parser uses a bidirectional parsing algorithm with two operations projection and non-projection to build the dependency tree. The performance of the proposed parser was evaluated based on the test data sentences. He reported that the system achieves a labelled attachment score of 71.63%, 59.86% and 67.74% for Hindi, Telugu and Bangla respectively on the treebank with fine-grained dependency labels. Based on the coarse-grained labels the dependency parser achieved 76.90%, 70.34% and 65.01% accuracies respectively.

xvi) Pradeep Kumar Das proposed a generative approach to the computation of basic verbal-strings in Hindi in 2008 [92]. He described a way to examine the possibility of developing a computational parser for verb morphology in Hindi that would generate correct verbal stems for different kinds of tense and aspects.

xvii) A parsing criteriafor Assamese text was described by Navanath Saharia, Utpal Sharma and Jugal Kalita in 2011 [93]. They described the practical analysis of Assamese sentences from a computational perspective rather than from linguistics perspective. This approach can be used to parse the simple sentences with multiple noun, adjective and adverb clauses.

xviii) An attempt to study the semantic relation of Causality or Cause-Effect was proposed by Sobha Lalitha Devi and Menaka S in 2011 [94]. They also described how semantic relation of Causality is marked in Tamil, how the causal markers in Tamil manifest in texts, their syntactic and semantic properties and how this information can be represented so as to handle causal information and reasoning.

xix) Akshar Bharati, Mridul Gupta, Vineet Yadav, Karthik Gali and Dipti Misra Sharma proposed a simple parser for Indian Languages in a dependency framework in 2009 [95]. They described a parser which uses a grammar driven approach to annotate dependency relations in both inter and intra chunk at an intermediary level. They described a grammar oriented model that makes use of linguistic features to identify relations. The proposed parser was modelled based on Paninian grammatical approach which provides a dependency grammar framework. They also compared the proposed parser performance against the previous similar attempts and reported its efficiency.

xx) An approach to expedite the process of manual annotation of a Hindi dependency Treebank was described by Gupta, Vineet Yadav, Samar Husain and Dipti Misra

Sharma [96]. They proposed a way by which consistency among a set of manual annotators could be improved. They have also showed that their setup can useful for evaluating, when an inexperienced annotator is ready to start participating in the production of the treebank. The performance of the system was evaluated on sample sets of data.

xxi) An unlabelled dependency parsing on graph based method for building multilingual weakly supervised dependency parsers for Hindi language was proposed by Jagadeesh Gorla, Anil Kumar Singh, Rajeev Sangal, Karthik Gali, Samar Husain, and Sriram Venkatapathy [78]. The system consists of two steps where the first step involves marking the chunks and the chunk heads of a given sentence and then identifying the intra-chunk dependency relations. The second step involves learning to identify the inter-chunk dependency relations. They reported that the system achieved a precision of 80.83% for sentences of size less than 10 words and 66.71% overall. They concluded that the result obtained was significantly better than the baseline in which random initialization is used.

2.6 MT APPROACHES AND SURVEY FOR INDIAN LANGUAGES

The term MT is a standard name for computerized systems responsible for the production of translations from one natural language into another with or without human assistance. It is a sub-field of computational linguistics that investigates the use of computer software to translate text or speech from one natural language to another. MT systems can be designed either specifically for two particular languages called bilingual system, or for more than a single pair of languages called multilingualsystems. Bilingual system may be either unidirectional, from one SL into one TL, or bidirectional. Multilingual systems are usually designed to be bidirectional but most bilingual systems are unidirectional. MT methodologies are commonly categorized as direct, transfer, and IL. The methodologies differ in the depth of analysis of the source language and the extent to which they attempt to reach a language independent representation of meaning or intent between the source and target languages. Barriers in good quality MT output can be attributed to ambiguity in natural languages. Ambiguity can be classified into two types: structurally ambiguous and lexically ambiguous.

Many attempts are being made all over the world to develop MT systems for various languages using rule based as well as statistical based approaches. Development of a well fledged Bilingual Machine Translation (BMT) system for any two natural languages with limited electronic resources and tools is a challenging and demanding task. In order to achieve a reasonable translation quality in open source tasks, corpus based MT approaches require large amounts of parallel corpus which are not always available, especially for less resourced language pairs.On the other hand the rule based MT process is extremely time consuming, difficult and failed to analyze accurately a large corpus of unrestricted text. This section gives a brief description about the various approaches and major MT developments in India.

2.6.1 History of MT

The major changeovers in MT system are as shown in Fig. 2.5. The theory of MT predates computers, with philosophers ’Leibniz and Descartes’ ideas of using code to relate words between languages in the seventeenth century [97]. The early 1930s saw the first patents for ’translating machines’. Georges Artsrouni was issued his patent in France in July 1933. He developed a device, which he called as ’cerveau mécanique’ (mechanical brain) that could translate between languages using four components: memory, a keyboard for input, a search method and an output mechanism. The search method was basically a dictionary look-up in the memory and therefore Hutchins is reluctant to call it a translation system. The proposal of Russian Petr Petrovich Troyanskii patented in September 1933 even bears a resemblance to the Apertium system, using a bilingual dictionary and a three-staged process, i.e. first a native speaking human editor of the SL pre-processed the text, then the machine performed the translation, and finally a native-speaking human editor of the TL post-edited the text [97,98].

After the birth of computers (ENIAC - Electrical Numerical Integrator And Calculator) in 1947, research began on using computers as aids for translating natural languages [99]. The first public demonstration of MT in the Georgetown-IBM experiment which proved deceptionally promising, encouraging financing of further research in the field. In 1949, Weaver wrote a memorandum, putting forward various proposals, based on the wartime successes in code breaking, the developments in information theory and speculations about universal principles underlying natural languages.The decade of optimism from 1954-1966, researchers encountered many predictions of imminent ’breakthroughs’. In the year 1966, Automated Language Processing Advisory Committee (ALPAC) report was submitted, which says for ’semantic barriers’ there is no straightforward solutions. The ALPAC report committee could not find any "pressing need for MT" nor "an unfulfilled need for translation [102]". This report brought MT research to its knees, suspending virtually all research in the USA while some research continued in Canada, France and Germany [99]. Since after the ALPAC report MT was almost took off from 1966-1980. In the year 1988, Georgetown-IBM experiment launched “IBM CANDIDE System”, where over 60 Russian sentences were translated smoothly into English using 6 rules and a bilingual dictionary consisting of 250 Russian words, with rule-signs assigned to words with more than one meaning. Although Professor Leon Dostert cautioned that this experimental demonstration was only a scientific sample, or "a Kitty Hawk of electronic translation”[100].

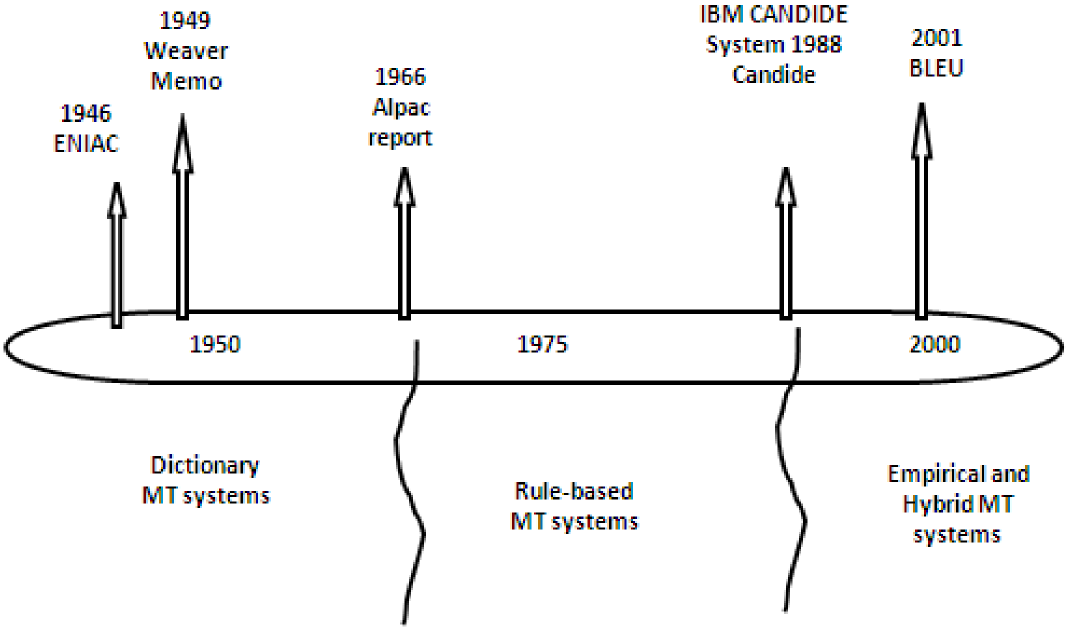

Fig. 2.5: The major changeovers in MT System

A wide variety of MT systems emerged after 1980 from various countries while research continued on more advanced methods and techniques. Those systems mostly comprised of indirect translations or used an IL as its intermediate. In the 1990s Statistical Machine Translation (SMT) and what is now known as ExampleBased Machine Translation (EBMT) saw the light of day [101]. At this time the focus of MT began to shift somewhat from pure research to practical application using hybrid approach. Moving towards the change of the millennium, MT became more readily available to individuals via online services and software for their Personal Computers (PCs).

2.6.2 MT Approaches

Generally MT is classified into seven broad categories such as: rule based, statistical based, hybrid based, example based, knowledge-based, principle-based and online interactive based methods. The first three MT approaches are most widely used and earliest methods. Literature shows that there are certain fruitful attempts using all these approaches for the development of English to Indian languages as well as Indian languages to Indian languages. At present most of the MT related research is based on statistical and example based approach. Fig.2.6 shows the classification of MT in NLP.

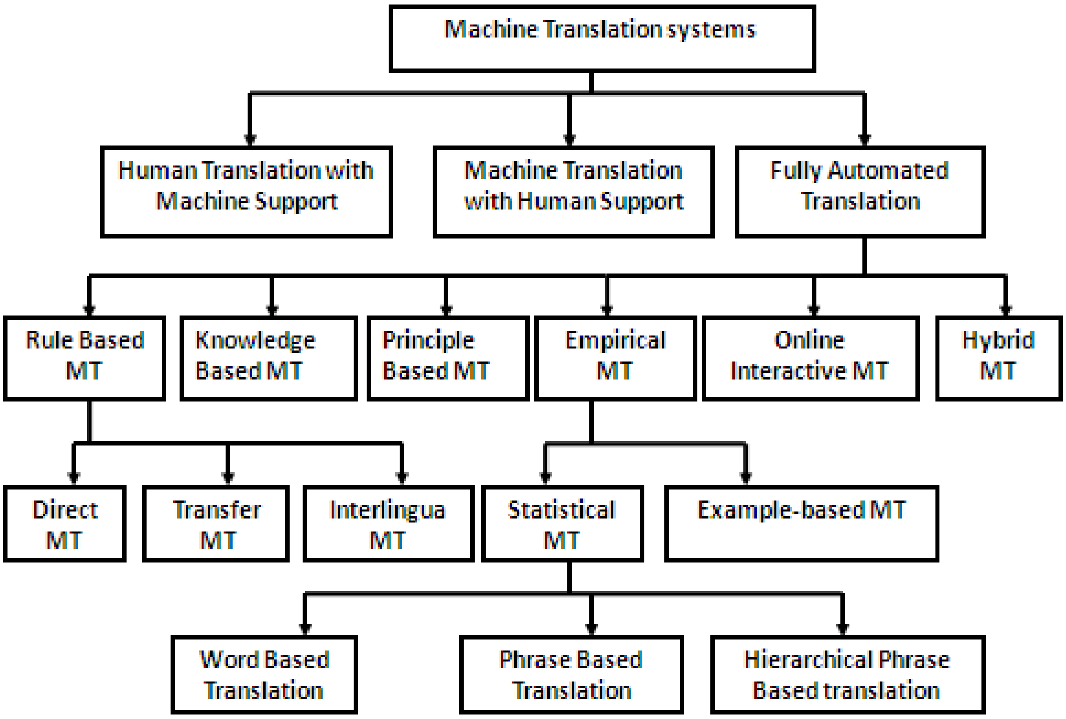

Fig.2.6: Classification of MT System

2.6.2.1 Rule Based Approach

In the field of MT, rule based approach is the first strategy ever developed. A RBMT system consists of collection of rules called grammar rules, bilingual or multilingual lexicon and software programs to process the rules. However, building RBMT systems entails a huge human effort to code all the linguistic resources such, source side part-of-speech taggers and syntactic parser, bilingual dictionaries, source to target transliteration, target language morphological generator, structural transfer and reordering rules. But a RBMT system is always extensible and maintainable. Rules play major role in various stages of translation such as syntactic processing, semantic interpretation, and contextual processing of language. Generally rules are written with linguistic knowledge gathered from linguists. Transfer-based MT, ILMT and dictionary-based MT are the three different approaches which come under rulebased machine translation category. In case of English to Indian languages and Indian language to Indian language MT system, there are certain fruitful attempts with all the four approaches. The main idea behind these rule based approaches as follows:

2.6.2.1.1Direct Translation

In the direct translation method the source language text was analyzed structurally up to morphological level and is designed for a specific source and target language pair[103,104]. The performance of direct MT system depends on the quality and quantity of the source-target language dictionaries, morphological analysis, text processing software and word-by-word translation with minor grammatical adjustments on word order and morphology.

2.6.2.1.2 Interlingua Based Translation

The next progress in the development of MT system is the IL approach where translation is performed by first representing the source language text into an intermediary (semantic) form called IL. The advantage of this approach is that IL is a language independent representation from which translations can be generated to different target languages. Thus the translation consists of two stages, where the source language is first converted into the IL form and then translation from the IL to the target language. The main advantage of this IL approach is that the analyzer of parser for the source language is independent of the generator for the target language. There are two main drawbacks in the IL approach. The first disadvantage is, its difficulty to define the IL. The second disadvantage is IL does not take the advantage of similarities between languages such as translation between Dravidian languages. The advantage of IL is it is economical in situations where translation among multiple languages is involved [105].

Starting with the shallowest level at the bottom, direct transfer is made at the word level. Moving upward through syntactic and semantic transfer approaches, the translation occurs on representations of the source sentence structure and meaning, respectively. Finally, at the interlingual level, the notion of transfer is replaced with a single underlying representation called the IL. IL represents both the source and target texts simultaneously. Moving up the triangle reduces the amount of work required to traverse the gap between languages, at the cost of increasing the required amount of analysis and synthesis.

2.6.2.1.3 Transfer Based Translation

Because of the disadvantage of IL approach, a better rule based translation approach was discovered called transfer approach. Recently many research groups have been using this third approach for their MT system both in India and abroad. On the basis of the structural differences between the source and target language a transfer system can be broken down into three different stages such as: i)Analysis, ii)Transfer and iii)Generation. In the first stage the source language parser is used to produce the syntactic representation of a source language sentence. In the next stage the result of first stage is converted into equivalent target language-oriented representations. In the final step of this translation approach, a target language morphological analyzer is used to generate the final target language texts.

2.6.2.2 Statistical Based Approach

Statistical approach comes under Empirical MT system which relies on large parallel aligned corpora. Statistical machine translation is a data oriented statistical framework for translating text from one natural language to another based on the knowledge extracted from bilingual corpus. Translations are generated on the basis of statistical models whose parameters are derived from the analysis of bilingual text corpora. In statistical based MT, bilingual or multilingual text corpora of source and target language or languages are required. Using any supervised or unsupervised statistical machine learning algorithm is used to build statistical tables from the corpora and this process is called the learning or training [77]. The statistical tables consist of statistical information such as the characteristics of well formed sentences, and the correlation between the languages. During translation, the collected statistical information is used to find the best translation for the input sentences and this translation step is called the decoding process. There are three different statistical approaches in MT called Word-based Translation, Phrase-based Translation and Hierarchical phrase based model.

The idea behind statistical machine translation comes from information theory. A document is translated according to the probability distribution function indicated by *p(e|f)*. Probability of translate a sentence *f* in the source language *F* (for example, English) to a sentence *e* in the target language *E* (for example, Kannada).

The problem of modelling the probability distribution *p(e|f)* has been approached in a number of ways. One intuitive approach is to apply Bayes theorem. That is, if *p(f|e)* and *p(e)* indicates translation model and language model respectively, then probability distribution *p(e|f)* ∞ *p(f|e)p(e)*. The translation model *p(f|e)* is the probability that the source sentence is the translation of the target sentence or the way sentences in *E* get converted to sentences in *F*. The language model *p(e)* is the probability of seeing that target language string or the kind of sentences that are likely in the language *E*. This decomposition is attractive as it splits the problem into two sub problems. Finding the best translation $\tilde{e}$is done by picking up the one that gives the highest probability as shown in equation 2.3.

$$\tilde{e} = \arg\max_{e \in e^*} p(e \mid f) = \arg\max_{e \in e^*} p(f \mid e) p(e) \qquad (2.3)$$

Even though phrase based models have emerged as most successful method for SMT they do not handle syntax in a natural way. Reordering of phrases during translation is typically managed by distortion models in SMT. But this reordering process is entirely not satisfactory especially for language pairs that differ a lot in terms of word-order. In the proposed project the problem of structural differences between source and target languages are successfully overcome with a reordering task. We have also proven that with the use of morphological information, especially for morphologically rich language like Kannada, the training data size can be much reduced with an improvement in performance.

2.6.2.2.1 Word Based Translation

As the name suggests, the words in an input sentence are translated into word by word individually and these words are finally arranged in a specific way to get the target sentence. The alignment between the words in the input and output sentences normally follows certain patterns in word based translation. This approach is the very first attempt in the statistical based MT system which is comparatively simple and efficient. The main disadvantage of this system is the oversimplified word by word translation of sentences which may reduce the performance of the translation system.

2.6.2.2.2 Phrase Based Translation

A more accurate SMT approach called phrase-based translation (Koehn et al., 2003) was introduced, where each source and target sentences are divided into separate phrases instead of words before translation. The alignment between the phrases in the input and output sentences normally follows certain patterns which is very similar to word based translation. Even though the phrase based model result a better performance than the word based translation, they did not improve the model of sentence order patterns. The alignment model is based on flat reordering patterns and experiments shows that this reordering technique may perform well with local phrase orders, but not as well with long sentences and complex orders.

2.6.2.2.3 Hierarchical Phrase Based model

By considering the drawback of previous two methods, Chiang (2005) developed a more sophisticated SMT approach called hierarchical phrase based model. The advantage of this approach is that hierarchical phrases have recursive structures instead of simple phrases. This higher level of abstraction approach further improved the accuracy of the SMT system.

2.6.2.3 Hybrid Based Translation

By taking the advantages of both statistical and rule-based translation methodologies a new approach is developed called hybrid based approach, which has proven a better efficiency in the area of MT system. At present several governmental and private based MT sectors using this hybrid based approach to develop translation from source to target language, which is based both rules and statistics. The hybrid approach can be used in number of different ways. In some case, translations are performed in the first stage using rules based approach followed by adjust or correct the output using statistical information. On the other way, rules are used to pre-process the input data as well as post-process the statistical output of a statistical based translation system. This technique is better than the previous and has more power, flexibility and control in translation.

Hybrid approaches integrating more than one MT paradigm are receiving increasing attention. The METIS-II MT system is an example of hybridization around the EBMT framework; it avoids the usual need for parallel corpora by using a bilingual dictionary (similar to that found in most RBMT systems) and a monolingual corpus in the TL [106]. An example of hybridization around the rule-based paradigm is given by Oepen. It integrate statistical methods within an RBMT system to choose the best translation from a set of competing hypotheses (translations) generated using rule-based methods [107].

In SMT, Koehn and Hoang integrate additional annotations at the word-level into the translation models in order to better learn some aspects of the translation that are best explained on a morphological, syntactic or semantic level [108]. Hybridization around the statistical approach to MT is provided by Groves and Way; they combine both corpus-based methods into a single MT system by using phrases (sub-sentential chunks) both from EBMT and SMT into an SMT system [109]. A different hybridization happens when an RBMT system and an SMT system are used in a cascade; Simard propose an approach, analogous to that by Dugast, which consists of using an SMT system as an automatic post-editor of the translations produced by an RBMT system [110,111].

2.6.2.4 Example based translation

Example based translation approach is based on analogical reasoning between two translation examples, proposed by Makoto Nagao in 1984. At run time an example based translation is characterized by its use of a bilingual corpus as its main knowledge base. The example based approach comes under Empirical MT system which relies on large parallel aligned corpora.

Example-based translation is essentially translation by analogy. An EBMT system is given a set of sentences in the source language (from which one is translating) and their corresponding translations in the target language, and uses those examples to translate other, similar source-language sentences into the target language. The basic premise is that, if a previously translated sentence occurs again, the same translation is likely to be correct again. EBMT systems are attractive in that they require a minimum of prior knowledge and are therefore quickly adaptable to many language pairs.

A restricted form of example-based translation is available commercially, known as a translation memory. In a translation memory, as the user translates text, the translations are added to a database, and when the same sentence occurs again, the previous translation is inserted into the translated document. This saves the user the effort of re-translating that sentence, and is particularly effective when translating a new revision of a previously-translated document.

More advanced translation memory systems will also return close but inexact matches on the assumption that editing the translation of the close match will take less time than generating a translation from scratch. ALEPH, wEBMT, English to Turkish, English to Japanese, English to Sanskrit and PanEBMT are some of the example based MT systems.

2.6.2.5Knowledge-Based MT

Knowledge-Based MT (KBMT) is characterized by a heavy emphasis on functionally complete understanding of the source text prior to the translation to the target text. KBMT does not require total understanding, but assumes that an interpretation engine can achieve successful translation into several languages. KBMT is implemented on the IL architecture; it differs from other interlingual techniques by the depth with which it analyzes the source language and its reliance on explicit knowledge of the world.

KBMT must be supported by world knowledge and by linguistic semantic knowledge about meanings of words and their combinations. Thus, a specific language is needed to represent the meaning of languages. Once the source language is analyzed, it will run through the augmenter. It is the Knowledgebase that converts the source representation into an appropriate target representation before synthesizing into the target sentence.

KBMT systems provide high quality translations. However, they are quite expensive to produce due to the large amount of knowledge needed to accurately represent sentences in different languages. English-Vientnamese MT system is one of the examples of KBMTS.

2.6.2.6 Principle-Based MT

Principle-Based MT (PBMT) Systems employ parsing methods based on the Principles & Parameters Theory of Chomsky‘s Generative Grammar. The parser generates a detailed syntactic structure that contains lexical, phrasal, grammatical, and thematic information. It also focuses on robustness, language-neutral representations, and deep linguistic analyses.

In the PBMT, the grammar is thought of as a set of language-independent, interactive well-formed principles and a set of language-dependent parameters. Thus, for a system that uses n languages, n parameter modules and one principles module are needed. Thus, it is well suited for use with the interlingual architecture.

PBMT parsing methods differ from the rule-based approaches. Although efficient in many circumstances, they have the drawback of language-dependence and increase exponentially in rules if one is using a multilingual translation system. It provides broad coverage of many linguistic phenomena, but lacks the deep knowledge about the translation domain that KBMT and EBMT systems employ. Another drawback of current PBMT systems is the lack of the most efficient method for applying the different principles. UNITRAN is one of the examples of Principle Based Machine Translation (PBMT) system.

2.6.2.7 Online Interactive Systems

In this interactive translation system user is allowed to suggest the translator to the correct translation in online. The advantage of this approach is that when the context of a word is unclear such that there are many possible meanings exists for particular word. In such cases the structural ambiguity can be easily solved with the interpretation of user.

2.6.3 Major MT Developments in India: A Literature Survey

A first public Russian to English [112]MT system was presented at the University of Georgetown in 1954 with a vocabulary size of around 250 words. Since then, many research projects were devoted to MT. However, as the complexity of the linguistic phenomena involved in the translation process together with the computational limitations of the time were made apparent, enthusiasm faded out quickly. Also the results of two negative reports namely ‘Bar-Hillel’ and ‘AL- PAC’ had a dramatic impact on MT research by that decade.

During the 1970s, the focus of MT activity switched from the United States to Canada and to Europe, especially due to the growing demands for translations within their multicultural societies. ‘Mateo’, a fully-automatic system translating weather forecasts had a great success in Canada. Meanwhile, the European Commission installed a French–English MT system called ‘Systran’. Other research projects, such as ‘Eurotra’, ‘Ariane’ and ‘Susy’, broadened the scope of MT objectives and techniques. The rule-based approaches emerged as the right way towards successful MT quality. Throughout the 1980s many different types of MT systems appeared and the most prevalent being those using an intermediate semantic language such as the IL approach.

Lately, various researchers have shown better translation quality with the use of phrase translation. Most competitive statistical machine translation systems such as the CMU, IBM, ISI, and Google etc. used phrase-based systems and came out with good results.

In the early 1990s, the progress made by the application of statistical methods to speech recognition introduced by IBM researchers was purely-statistical machine translation models [112]. The drastic increment in computational power and the increasing availability of written translated texts allowed the development of statistical and other corpus-based MT approaches. Many academic tools turned into useful commercial translation products, and several translation engines were quickly offered in the World Wide Web.

Today, there is a growing demand for high-quality automatic translation. Almost all the research community has moved towards corpus-based techniques, which have systematically outperformed traditional knowledge-based techniques in most performance comparisons. Every year more research groups embark on SMT experimentation and a regained optimism as regards to future progress seems to be shared among the community.

Machine translation is an emerging research area in NLP for Indian languages, which stared more than a decade ago. There are number of attempts in MT for English to Indian languages and Indian languages to Indian languages using different approaches. Literature shows that the earliest published work was undertaken by Chakraborty in 1966 [103]. Many governmental, private sectors as well as individuals are actively involving in the development of MT system and have already generated some reasonable MT system. Some of these MT systems are in advanced prototype or technology transfer stage, and the rest having been newly initiated.The main development in Indian language MT system are as follows:

2.6.3.1 ANGLABHARTI by Indian Institute of Technology (IIT), Kanpur (1991)

ANGLABHARTI is a multilingual machine aided translation project on translation from English to Indian languages primarily Hindi, which is based pattern directed approach [113,114,115,112,117,116]. The strategy in this MT system is better than the transfer approach and lies below the IL approach. In the first stage a pattern directed parsing is performed on the source language English and generates a `pseudo-target' which is applicable to a set of Indian languages. Word sense ambiguity in the source language sentence is also resolved by using a number of semantic tags. In order to transform the pseudo target language into their corresponding target language, the system uses a separate text generator module. After correcting all ill formed target sentences, a post editing package is used make the final corrections. Even though it is a general purpose system, at present it has been applied mainly in the domain of public health. ANGLABHARTI system is currently implemented from English-to Hindi translation called AnglaHindi which is web- enabled (http:/anglahindi.iitk.ac.in) and obtained good domain specific results for health campaign and have been successfully translated many pamphlets and medical booklets. At present, further research work is going on to extend this approach for English to Telugu/Tamil translation. The project is primarily based at IIT-Kanpur, in collaboration with ER&DCI, Noida, and has been funded by TDIL. Professor RMK Sinha, IIT, Kanpur is leading this MT project.

2.6.3.2 ANGLABHARTI –II by IIT, Kanpur (2004)

The disadvantages of the previous system are solved by introducing the ANGLABHARTI –II MT architecture system [118]. The different approach called a Generalized Example-Base (GEB) for hybridization besides a Raw Example-Base (REB) is used to improve the performance of the translation. Comparing with the previous approach, the system first attempts a match in REB and GEB before invoking the rule-base at the time of actual usage. Automated pre-editing and paraphrasing step are the further improvement in the proposed new translation approach. The system is designed in a way that various submodules are pipelined in order to achieve more accuracy and robustness.

At present the AnglaBharti technology has been transfered under AnglaBharti Mission into eight different sectors across the country [117]. The main intention of this bifurcation is to develop Machine Aided Translation (MAT) systems for English to twelve Indian regional languages. This include MT from English to Marathi & Konkani (IIT Mumbai), English to Asamiya and Manipuri (IIT Guwahati), English to Bangla (CDAC Kolkata), English to Urdu, Sindhi & Kashmiri (CDAC-GIST group) Pune, English to Malyalam (CDAC Thiruvananthpuram), English to Punjabi (Thapar Institute of Engineering and Technology-TIET, Patiala), English to Sanskrit Jawaharlal Nehru University –JNU, New Delhi), and English to Oriya (Utkal University Bhuvaneshwar).

2.6.3.3 ANUBHARATI by IIT, Kanpur (1995)

Anubharati is recently started MT system aimed to translate from Hindi to English language [113,114,115,117,116]. Similar to Anglabharti MT system, anubharti is also based on machine aided translation in which a variation of example-based approach called template or hybrid called HEBM is used. Literature shows that a prototype version of the MT system has been developed and the project is being extended for developing complete system. HEBMT approach takes the advantages of pattern and example based approaches by combining the essentials of these methods. One more added advantage of this Anubharti system is that, it provides a generic model for translation that is suitable for translation between any two Indian languages pair with minor addition of modules.

2.6.3.4 ANUBHARATI-II by IIT, Kanpur (2004)

ANUBHARATI-II is a revised version of the ANUBHARATI, which overcomes most of the drawbacks of the earlier architecture with varying degree of hybridization of different paradigms [117]. The main intention of this system is to develop Hindi to any other Indian languages, with a generalized hierarchical example-base approach. However both ANGLABHARTI and ANUBHARTI systems did not produced the expected results, both system have been successfully implemented with good results. Professor RMK Sinha, IIT, Kanpur is leading this MT project.

2.6.3.5 Anusaaraka by IIT, Kanpur and University of Hyderabad

To utilize the close similarity among Indian languages for MT, another translation system was introduced called Anusaaraka [113,117] which is based on the principles of Paninian Grammar (PG). Anusaaraka is a machine aided translation system which is also used on language access between these languages. At present this system is applied for children's stories and an Alpha version of the system has been already developed for language accessors from five regional languages such as Punjabi, Bengali, Telugu, Kannada and Marathi into Hindi. The Anusaaraka MT approach mainly consist of two modules [112,119]: the first module is called Core Anusaarakawhich is based on language knowledge and the second one is a domain specific module which is based statistical knowledge, world knowledge, etc. That is the criteria behind the Anusaaraka are different from other systems in such a way that the total load is divided into parts. The machine carried out the language-based analysis of the text and the remaining work such as knowledge-based analysis or interpretation is performed by the reader. The Anusaaraka project was funded by TDIL, it started at IIT Kanpur, and later shifted mainly to the Centre for Applied Linguistics and Translation Studies (CALTS), Department of Humanities and Social Sciences, University of Hyderabad. At present the Language Technology Research Centre (LTRC) at IIIT Hyderabad is developing an English to Hindi MT system using the architecture of Anusaaraka approach. This Anusaaraka project is developing under the supervision of Prof. Rajeev Sangal and Prof. G U Rao.

2.6.3.6 Anusaaraka System from English to Hindi

Anusaaraka system from English to Hindi preserves the basic principles of information preservation and load distribution of original Anusaaraka [112,119]. To analyze the source text, it uses modified version of XTAG based super tagger and light dependency analyzer which is developed at University of Pennsylvania. The advantage of this system is that after the completion of source text analysis, user may read the output and also the user can always move to a simpler output if the system produces a wrong output or fails to produce the output.

2.6.3.7 MaTra (2004)

MaTra is English to Indian languages (at present Hindi) Human-Assisted translation system based on a transfer approach using a frame-like structured representation and resolves the ambiguities using rule-bases and heuristics [113,117,112]. MaTra is an innovative system, which provides an intuitive GUI, where user can visually inspect the analysis of the system and can provide disambiguation information to produce a single correct translation. Even though the MaTra system aimed to general purpose system, it has been applied mainly in the domains of news, annual reports and technical phrases. MaTra is an ongoing project and currently the system is able to translate domain specific simple sentences and work is on towards to cover the other types of sentences. Natural Language group of the Knowledge Based Computer Systems (KBCS) division at the National Centre for Software Technology (NCST), Mumbai (currently CDAC, Mumbai) is undertaken to develop MaTra system which is funded by TDIL.

2.6.3.8 MANTRA by Centre for Development of Advanced Computing, Bangalore (1999)

Mantra MT system is intended to perform translation for the domains of gazette notifications pertaining to government appointments, parliament proceeding summaries, between English to Indian languages as well as Indian languages to English, where source and target language grammars represented using Lexicalized Tree Adjoining Grammar (LTAG) formalism [113,117]. The added advantage of this system is that the system can also preserve the formating of input Word documents across the translation. After the successful development of MANTRA-Rajyasabha, language pairs like Hindi-English and

Hindi-Bengali translation are started already using Mantra approach. The Mantra project is developing under the supervision of Dr Hemant Darbari, funded by TDIL, and later by the Department of Official Languages underMinistry of Home Affairs ofGovernment of India.

2.6.3.9 UCSG-based English-Kannada MT by University of Hyderabad

Using the UCSG formalism, the Computer and Information Sciences Department at the University of Hyderabad under the supervision of Prof. K Narayana Murthy, developed a domain specific English-Kannada MT system [113,117,112]. This UCSG-based system is based on transfer-based approach, and has been applied to the translation of government circulars. The system works at sentence level and requires post-editing. At its first step of translation, the source (English) sentence is analysed and parsed using UCSG parser (developed by Dr. K. Narayana Murthy), and then using translation rules, English-Kannada bilingual dictionary and network based Kannada Morphological Generator (developed by Dr. K. Narayana Murthy), the system translate into Kannada language. This project has been funded by government of Karnataka and work is going to improve the performance of the system. Later the same approach was applied for English-Telugu translation.

2.6.3.10 UNL-based MT between English, Hindi and Marathi by IIT, Mumbai

Universal Networking Language (UNL) based MT between English, Hindi and Marathi is based on IL approach [113,117,112]. Under the supervision of Prof. Pushpak Bhattacharya, IIT Bombay is the Indian participant in UNL, which is an international project of the United Nations University, aimed to develop an IL for all major human languages in the world. In the UNL based MT, the knowledge of the source language is captured or enconverted into UNL form and deconverted from UNL to the target language like Hindi and Marathi. The source language information is represented into sentence by sentence which is later converted into a hypergraph having concepts as nodes and relations as directed arcs [105]. The document knowledge is expressed in three dimensions as: word knowledge, conceptual knowledge and attribute labels.

2.6.3.11 Tamil-Hindi Anusaaraka MT

KB Chandrasekhar Research Centre of the Anna University at Chennai is active in the area of Tamil NLP. A Tamil-Hindi language accessor has been built using the Anusaaraka formalism[113,117,112]. The group has developed a Tamil-Hindi machine aided translation system under the supervision of Prof. CN Krishnan, and has a performance of 75%.

2.6.3.12 English-Tamil machine Aided Translation system

Recently, the NLP group also developed a prototype ofEnglish-Tamil Human Aided Machine Translation (HAMT) System [112,121]. The system mainly consists of three major components: English morphological analyzer, mapping unit the Tamil language morphological generator.

2.6.3.13 SHIVA MT System for English to Hindi

This project is developed jointly by Indian Institute of Science, Bangalore, and International Institute of Information Technology, Hyderabad in collaboration with Carnegie Mellon University USA based on Example-based approach [117,121]. An experimental system has been released for experiments, trials, and user feedback and is publicly available.

2.6.3.14 SHAKTI MT System for English to Hindi, Marathi and Telugu

This is a recently started project which is also developed jointly by Indian Institute of Science, Bangalore, and International Institute of Information Technology, Hyderabad in collaboration with Carnegie Mellon University USA [117,121]. The system follow a hybrid approach by combining both rule as well as statistical based approaches. An experimental system for English to Hindi, Marathi and Telugu has been is publicly available for experiments, trials, and user feedback.

2.6.3.15 Anuvadak English-Hindi MT

Anuvadak 5.0 English to Hindi software is a general-purpose tool developed by one of the private sector called Super Infosoft Pvt Ltd., Delhi under the supervision of Mrs. Anjali Rowchoudhury [113,117,112,121]. The system has inbuilt dictionaries in specific domains and support post-editing. If the corresponding target word is not present in the lexicon diction, the system has a facility to translate that source word into target. The system can run in Window family and a demonstration version of the system is publicly available.

2.6.3.16 English-Hindi Statistical MT

Statistical based English to Indian languages, mainly Hindi MT system is started by IBM India Research Lab at New Delhi, which uses the same approach as its existing work on other language [113,112].

2.6.3.17 English-Hindi MAT for news sentences

A rule based English to Hindi Machine Aided Translation system was developed by Jadavpur University, Kolkata under the supervision of Prof. Sivaji Bandyopadhyay [113]. The system uses transfer based approach and currently working on domain specific MT system for news sentences.

2.6.3.18 Hybrid MT system for English to Bengali

Under the supervision of Prof. Sivaji Bandyopadhyay, a hybrid based MT system for English to Bengali is developed at Jadavpur University Kolkata in 2004 [121]. The current version of the system works at the sentence level.

2.6.3.19 Hinglish MT system

In the year 2004, Prof. Sinha and Prof. Thakur developed a standatrd Hindi - English MT system called Hinglish by incorporating additional level to the existing AnglaBharti-II and AnuBharti-II systems [121]. The system produced satisfactory acceptable results in more than 90% of the cases, except the case with polysemous verbs.

2.6.3.20 English to (Hindi, Kannada, Tamil) and Kannada to Tamil language-pair EBMT system (2006)

Example based English to {Hindi, Kannada and Tamil} and Kannada to Tamil [121]MT system were developed by Balajapally *et al*. (2006). A set of bilingual dictionaries comprising of sentence dictionary, phrases-dictionary, words-dictionary and phonetic-dictionary of parallel corpora of sentence, phrases, words and phonetic mappings of words is used for the MT. A corpus size of 75,000 most commonly used English-{Hindi, Kannada and Tamil} sentences pairs are used for MT.

2.6.3.21 Punjabi to Hindi MT system (2007)

A direct word-to- word translation approach, a Punjabi to Hindi MT system was developed by Josan and Lehal at Punjabi University, Patiala and reported 92.8% accuracy [121]. In addition to the Punjabi-Hindi lexicon and morphological analysis, the system also consists modules that support word sense disambiguation, transliteration and post processing.

2.6.3.22 Machine translation System among Indian language – Sampark (2009)

Consortiums of institutions (including IIIT Hyderabad, University of Hyderabad, CDAC (Noida, Pune), Anna University, KBC, Chennai, IIT Kharagpur, IIT Kanpur, IISc Bangalore, IIIT Alahabad, Tamil University, Jadavpur University) started to develop MT systems among Indian languages called Sampark and have already released experimental systems for {Punjabi, Urdu, Tamil, Marathi} to Hindi and Tamil-Hindi in 2009 [121].

2.6.3.23 English to Bengali (ANUBAAD) and English to Hindi MT System by Jadavpur University

Using the transfer based approach Jadavpur University developed a domain specific translation of English news to Bengali called ANUBAAD and current system work at sentence level [117]. Also the University started to develop a translation system for English news headlines to Bengali using a semantics-example based approach. Using the same architecture, the university also developed a MT system for English – Hindi and the system works currently at the simple sentence level. Recently the university also started to develop Indian languages (Bengali, Manipuri) to English MT system. These translation systems are developing under the supervision of Prof. Sivaji Bandyopadhyay. The university uses these translation systems for guiding students and researchers who work in the MT area.

2.6.3.24 Oriya MT System (OMTrans) by Utkal University, Vanivihar

Utkal University, Bhuvaneshwar is working on a English-Oriya MT system OMTrans under the supervision of Prof. Sanghamitra Mohanty [117,112]. The OMT system consists of six different parts: Parser, Translator, OMT System, OMT Database, Disambiguator and the Software tools. The heart of the system is the OMT bilingual dictionary database. The OMT system was implemented for various types of simple as well as complex sentences. The system was tested with various types of sentences taken from schoolbooks.

2.6.3.25 English-Hindi EBMT system by IIT Delhi

The Department of Mathematics, IIT Delhi under the supervision of Professor Niladri Chatterjee developed example based English-Hindi MT system [117]. They have developed divergence algorithms for identifying the divergence for English to Hindi example based system and a systematic scheme for retrieval from the English-Hindi example base.

2.6.3.26 Machine Aided Translation by Centre for Development of Advanced Computing (CDAC), Noida

Using the Machine Aided Translation system approach a domain specific translation system for translating public health related sentences from English to Hindi was developed [112]. The system support advantage of post editing and reported 60% performance.

2.6.3.27 Hindi to Punjabi MT system (2009)

Goyal and Lehal of Punjabi University, Patiala developed a Hindi to Punjabi Machine translation system based on direct word-to-word translation approach [122,121]. The system consists of the following modules: pre-processing, word-to-word Hindi-Punjabi lexicon, morphological analysis, word sense disambiguation, transliteration and post processing. They have also developed an evaluation approach for Hindi to English translation system and have reported 95% accuracy. Still work has been carried out to achieve better system.

2.6.3.28 A Statistical Machine Translation Approach to Sinhala-Tamil Language (2011)

Ruvan Weerasinghe developed a Statistical Machine Translation Approach to Sinhala-Tamil Language Translation [123]. This work reports on SMT based translation performed between language pairs such as Sinhala- Tamil and the English-Sinhala pair. The experiments results show that current models perform significantly better for the Sinhala-Tamil pair than the English-Sinhala pair and proven that SMT system work better for languages that are not too distantly related to each other.

2.6.3.29 An Interactive Approach for English-Tamil MT System on the Web (2002)

Dr. Vasu Renganathan, University of Pennsylvania, developed an interactive approach for of English-Tamil MT System on the Web [120]. The system is based on a rule based approach, containing around five thousand words in lexicon and a n umber of transfer rules used for mapping English structures to Tamil structures. This is an interactive system such that users can update this system by adding more words into lexicon and rules into rule-base.

2.6.3.30 Translation system using pictorial knowledge representation (2010)

Samir Kr. Borgohain and Shivashankar B. Nair introduced a new MT approach for Pictorially Grounded Language (PGL) based on their pictorial knowledge [124]. In this approach, symbols of both the source and the target languages being grounded on a common set of images and animations. PGL is a graphic language andact as a conventional intermediate language representation. While preserving the inherent meanings of the source language, the translation mechanism can also be scalable into a larger set of languages. The translation system is implemented in such a way that, images and objects are tagged with both the source and target language equivalents, which makes the reverse translation much easier.

2.6.3.31 Rule Based Reordering and Morphological Processing For English-Malayalam Statistical Machine Translation (2009)

This is an attempt to develop a statistical based MT for English to Malayalam language by a set of MTech students under the guidance of Dr. Soman K P [125]. In this approach they showed that a SMT based system can be improved by incorporating the rule based reordering and morphological information of source and target languages.

2.6.3.32 Statistical Machine Translation using Joshua (2011)

A piloted SMT based English to Telugu MT system called “enTel” was developed by Anitha Nalluri and Vijayanand Kommaluri, based on Johns Hopkins University Open Source Architecture (JOSHUA) [126]. A Telugu parallel corpus from the Enabling Minority Language Engineering (EMILLE) developed by CIIL Mysore and English to Telugu Dictionary developed by Charles Philip Brown is considered for training the translation system.

2.6.3.33 Multilingual Book Reader

Transliteration, Word-to-Word Translation and Full-text Translation for Indian languages: The NLP team including Prashanth Balajapally, Phanindra Bandaru, Ganapathiraju, N. Balakrishnan and Raj Reddy introduced a multilingual book reader interface for DLI, which supports transliteration and good enough translation [127]. This is a simple, inexpensive tool which exploits the similarity between Indian languages. This tool can be useful for beginers who can understand their mother tongue or other Indian languages, but cannot read the script and average reader who has the domain expertise. This tool can be also be used for translating either the documents or the queries in a multilingual search purpose.

2.6.3.34 A Hybrid Approach to Example based Machine Translation for English to Indian Languages (2007)

Vamshi Ambati and Rohini U proposed a hybrid approach to EBMT for English to Indian languages which make use of SMT methods and minimal linguistic resources [128]. Currently work is going on to develop English to Hindi as well as other Indian language translation system based on manual and a statistical dictionary build from SMT tool using an example database consisting of source and target parallel sentences.

2.6.3.35 Statistical Machine Translation by Incorporating Syntactic and Morphological Processing

Ananthakrishnan Ramanathan, Pushpak Bhattacharyya, Jayprasad Hegde, Ritesh M. Shah, and Sasikumar M proposed a new idea to improve the performance of the SMT based MT by incorporating syntactic and morphological processing [129]. In this contest they proved that performance of a baseline phrase-based system can be substantially improved by: i) reordering the source (English) sentence as per target (Hindi) syntax, and (ii) using the suffixes of target (Hindi) words.

2.6.3.36 Prototype MT System from Text-To-Indian Sign Language (ISL)

This is a very different approach to MT which intended for dissemination of information to the deaf people in India proposed by Tirthankar Dasgupta, Sandipan Dandpat, and Anupam Basu [130]. At present a prototype version of English to Indian Sign Language is developed and the ISL syntax is represented based on Lexical Functional Grammar (LFG) formalism.

2.6.3.37 An Adaptable Frame based system for Dravidian language Processing (1999)

In the proposed work, a different approach that makes use of the karaka relations for sentence comprehension is used in the frame based translation system for Dravidian languages [131]. Two patterns directed application-oriented experiments are conducted and the same meaning representation technique is used in both the cases. In the first experiment translation is done from a free word order language to fixed word order one, where both source and destination are natural languages. But in the second experiment the target language is an artificial language with a rigid syntax. Even though there is a difference in generation of target sentence, the results obtained in both experiments are encouraging.

2.6.3.38English-Telugu T2T MT and Telugu-Tamil MT System (2004)

CALTS in collaboration with, IIIT, Hyderabad; Telugu University, Hyderabad; Osmania University, Hyderabad developed an English-Telugu as well as Telugu-Tamil Machine translation systems under the supervision of Prof. Rajeev sangal [132]. The English-Telugu system uses an English-Telugu machine aided translation lexicon of size

42000 and a wordform synthesizer for Telugu. Telugu-Tamil MT system is developed based on the available resources at CALTS such as: Telugu Morphological analyzer, Tamil generator, verb sense disambiguator and Telugu-Tamil machine aided translation dictionary. The performances of the systems are encouraging and handle source sentences of a variety of complexity.

2.6.3.39 Developing English-Urdu MTvia Hindi (2009)

R. Mahesh K. Sinha proposed a different strategy for deriving English to Urdu translation using English to Hindi MT system [133]. In the proposed method an English-Hindi lexical database is used to collect all possible Hindi words and phrases. These words and phrases are further augmented by including their morphological variations and attaching all possible postpositions. Urdu is structurally very close to Hindi and this augmented list is used to provide mapping from Hindi to Urdu. The advantage of this translation system is that the grammatical analysis of English provides all the necessary information needed for Hindi to Urdu mapping and no part of speech tagging, chunking or parsing of Hindi has been used for translation.

2.6.3.40 Bengali- Assamese automatic MT system- VAASAANUBAADA (2002)

Kommaluri Vijayanand, S Choudhury and Pranab Ratna proposed a automatic bilingual MT for Bengali to Assamese using example based approach [134]. They used a manually created aligned bilingual corpus by feeding the real examples using pseudo code. The quality of the translation is improved by preprocessing the longer input sentences and also the backtracking techniques. Since the grammatical structure of Bengali and Assamese languages are very similar and lexical word groups is required.

2.6.3.41 Phrase based English – Tamil Translation System by Concept Labelling using Translation Memory (2011)

Computational Engineering and Networking research centre of Amrita School of Engineering, Coimbatore, proposed a English – Tamil translation system. The system is based on phrase based approach by incorporating concept labelling using translation memory of parallel corpus [135]. The translation system consists of 50,000 English – Tamil parallel sentences, 5000 proverbs, and 1000 idioms and phrases, with a dictionary containing more than 2,00,000 technical words and 100,000 general words and has the accuracy of 70%.

2.6.3.42 Rule based Sentence Simplification for English to Tamil MT System (2011)

This work is aimed to improve the translation quality of the MT system by simplifying the complex input sentences for English to Tamil MT system [136]. In order to simplify the complex sentences based on connectives like relative pronouns, coordinating and subordinating conjunction a rule based technique is proposed. In this approach a complex sentence is expressed as the list of sub-sentences while the meaning remains unaltered. The simplification task can be used as a preprocessing tool for MT where the initial splitting is based on delimiters and then the simplification is based on connectives.

2.6.3.43 Manipuri-English Bidirectional Statistical Machine Translation Systems (2010)

Using morphology and dependency relations a Manipuri to English bidirectional statistical machine translation system is developed by Thoudam Doren Singh and Sivaji Bandyopadhyay [137]. The system uses a domain specific parallel corpus of 10350 sentences from news for training purpose and the system is tested with 500 sentences.

2.6.3.44 English to Dravidian Language Statistical Machine Translation System (2010)

Unnikrishnan P,Antony P J and Dr Soman K P proposed a SMT system for English to Kannada by incorporating syntactic and morphological information [138]. In order to increase the performance of the translation system, we have introduced a new approach in creating parallel corpus. The main ideas which we have implemented and proven very effective for English to Kannada SMT system are: (i) reordering the English source sentence according to Dravidian syntax, (ii) using the root suffix separation on both English and Dravidian words and iii) use of morphological information which substantially reduce the corpus size required for training the system. The results show that significant improvements are possible by incorporating syntactic and morphological information to the corpus. From the experiment we have found that the proposed translation system successfully works for almost all simple sentences in their twelve tense forms and their negatives forms.

2.6.3.45 Anuvadaksh

This system is an effort of English to Indian Language Machine Translation (EILMT) consortium [139]. Anuvadaksh is a system that allows translating the text from English to six other Indian languages i.e. Hindi, Urdu, Oriya, Bangla, Marathi, Tamil. Anuvadaksh being a consortium based project is having a hybrid approach, designed to work with the platform and technology independent modules.This system has been developed to facilitate the multi-lingual community, initially in the domain-specific expressions of Tourism, and subsequently it would foray into various other domains as well in a phase-wise manner. It integrates four MT Technologies:

1. Tree-Adjoining-Grammar (TAG) based MT.
2. Statistical based Machine translation.
3. Analyze and Generate rules (Anlagen) based MT.
4. Example based MT.

2.6.3.46 Google Translate

Google Translate is a free translation service that provides instant translations between 57 different languages. Google Translate generates a translation by looking for patterns in hundreds of millions of documents to help decide on the best translation. By detecting patterns in documents that have already been translated by human translators, Google Translate makes guesses as to what an appropriate translation should be. This process of seeking patterns in large amounts of text is called "statistical machine translation".

2.6.3.47 English to Assamese MT System

English to Assamese MT system is in progress [139]. The following activities are under progress in this direction:

- The graphical user interface of the MT system has been re-designed. It now allows display of Assamese text. Modifications have been made in the Java modules.

- The existing Susha encoding scheme has been used. In addition, a new Assamese font set has been created according to that of Susha font set. The system is now able to display properly consonants, vowels, and matras of Assamese characters properly.

- The mapping of Assamese keyboard with that of Roman has been worked out.

- The process of entering Assamese words (equivalent of English words) in the lexical database (nouns and verbs) is in progress.

The system developed basically a rule-based approach and relies on a bilingual English toAssamese dictionary. The dictionary-supported generation of Assamese text from Englishtext is a major stage in this MT. Each entry in the dictionary is supplied with inflectional information about the English lexeme and all of its Assamese equivalents. The dictionary is annotated for morphological, syntactic and partially semantic information. It can currently handle translation of simple sentences from English to Assamese. The Dictionary contains around 5000 root words. The system simply translates source language texts to the corresponding target language texts phrase to phrase by means of the Bilingual dictionary lookup.

2.6.3.48 Tamil University MT System

Tamil University, Tanjore, initiated a machine oriented translation from Russian-Tamil during 1983-1984 under the leadership of the Vice-Chancellor Dr. V.I Subramaniam [139]. It was taken up as an experimental project to study and compare Tamil with Russian in order to translate Russian scientific text into Tamil. . A team consisting of a linguist, a Russian language scholar and a computer scientist were entrusted to work on this project. During the preliminary survey, both Russian SL and Tamil were compared thoroughly for their style, syntax, morphological level etc.

2.6.3.49 Tamil - Malayalam MT System

Bharathidasan University, Tamilnadu is working on translation between languages belonging to the same family such as Tamil - Malayalam Translation [139]. The MT consists of the following module and these are under progress.

Lexical database- This will be a bilingual dictionary of root words. All the noun roots and verb roots are collected.

Suffix database- Inflectional suffixes, derivative suffixes, plural markers, tense markers,sariyai, case suffixes, relative participle markers, verbal participle markers etc will be compiled.

Morphological Analyzer- It is designed to analyze the constituents of the words. It willhelp to segment the words into stems, inflectional markers.

Syntactic Analyzer- The syntactic analyzer will find the syntactic category like VerbalPhrase, Noun Phrase, Participle Phrase etc. This will analyze the sentences in the sourcetext.

2.7 SUMMARY

This chapter presented a survey on different developments in computational linguistics tools and MT systems for Indian languages. Additionally this chapter describes briefly the different existing approaches that have been used to develop various computational linguistics tools and MT systems. From the survey it found that almost all existing computational linguistics tools and Indian languages MT projects were based on statistical and hybrid approach. Since Indian languages are morphologically rich in feature and agglutinative in nature, rule based approaches failed in many situation for developing fully fledged tools and MT system. Secondly the general benefits of statistical and hybrid approaches, encouraged the researchers to choose these approaches to develop the tools and MT systems for Indian languages.

2.8 PUBLICATIONS

1. Antony P J and Soman K P: “Machine Transliteration for Indian Languages: A Literature Survey”, International Journal of Scientific and Engineering Research – IJSER, November, 2011. Abstracted and indexed in DOAJ, Google Scholar, Sribd, EBSCO, ScirRte, Scirus, Scientific Commons, CiteSeer, BASE, Computer Science Bibliography, Docstoc, Qsensei.

2. Antony P J and Soman K P: “Parts of Speech Tagging for Indian Languages: A Literature Survey”, International journal on Computer Application –IJCA (0975 – 8887) Volume 34– No.8, November 2011, Published by: Foundation of Computer Science,Abstracted and indexed in DOAJ, Google Scholar, Informatics, ProQuest CSA Technology Research Database. Impact factor: 0.87.

3. Antony P J and Soman K P: “Computational Morphology and Natural Language Parsing for Indian languages: A Literature Survey”, International Journal of Computer Science & Engineering Technology (IJCSET), ISSN: 2229-3345, Vol. 3 No. 4, April 2012.

4. Antony P J and Soman K P: “Machine Translation Approaches and Survey for Indian Languages”, International Journal of Computational Linguistics and Chinese Language Processing (Accepted).