CHAPTER II

REVIEW OF LITERATURE

Searching for useful information has become a mandatory part in a person's life in this information era. Usage of CLIR has removed the restriction, in searching information limited to a single language, and has allowed the user to extend their search to other languages. With the advent of CLIR, people can now search and obtain information in their native language. Tamil-English Cross-Language Information retrieval is a field of science that is an amalgamation of several fields including Natural Language Processing, statistics and machine learning. Tamil is one of the classical languages spoken by more than 77 million people across the world (Krishnamurti, 2003). Owing to the popularity of this language, a CLTR system that uses Tamil queries to retrieve English documents will be very useful in WWW, and is the focus of this research work. This chapter presents works related to information retrieval with particular emphasis on Tamil language.

2.1. TAMIL LANGUAGE

Tamil is a Dravidian language spoken predominantly in Tamil Nadu and Srilanka (http://en.wikipedia.org/wiki/Tamil_language). Tamil is regarded as one of the four major literary languages of the Dravidian family (Tamil, Kannada, Telugu and Malayalam) and is one of the longest surviving classical languages in the world (Stein, 1977; Thomas, 1998). Tamil-Brahmi inscriptions from 500BC have been found in Adichanallur (Subramanian, 2005) and 2,200-year-old Tamil-Brahmi inscriptions have been found in Samanamalai (Subramanian, 2012).

It has been described as "the only language of contemporary India which is recognizably continuous with a classical past" (Kamil, 1973). The variety and quality of classical Tamil literature has led to it being described as "one of the great classical traditions and literatures of the world" (George, 2015). Tamil literature has existed for over 2000 years (Kamil, 1992). The earliest period of Tamil literature, Sangam literature, is dated from sangam 300 BC- 300 AD (Abraham, 2003; Definitive Editions of Ancient Tamil Works, 2008).

It has the oldest extant literature amongst other Dravidian languages (Stein, 1977). The earliest epigraphic records found on rock edicts and hero stones date from around the 3rd century BC (Maloney, 1970; Subramaniam, 2011). About 60,000 of the one-lakh of the epigraphical inscriptions, found by the Archaeological Survey of India, are in Tamil Nadu and of them, only 5% are in languages other than Tamil (http://www.thehindu.com/2005/11/22/ stories/2005112215970400.htm, 2005).

Tamil language inscriptions written in Brahmi script have been discovered in Sri Lanka and on trade goods in Thailand and Egypt (http://www.thehindu.com/ todays-paper/tamil-brahmi-script-in-egypt/article1952611.ece,2007;Iravatham,2010). The two earliest manuscripts from India (The I.A.S. Tamil Medical Manuscript Collection, 2012; Saiva Manuscript in Pondicherry, 2012), acknowledged and registered by UNESCO Memory of the World register in 1997 and 2005, were in Tamil (Memory of the World Register: India, 2012).

In 1578, Portuguese Christian Missionaries published a Tamil prayer book in old Tamil script named 'Thambiraan Vanakkam', thus making Tamil the first Indian language to be printed and published (Madhavan, 2010). Tamil Lexicon, published by the University of Madras, is the first among the dictionaries published in any Indian language (Kolappan, 2014). Tamil is used as a sacred language of Ayyavazhi and in Tamil Hindu traditions of Shaivism and Vaishnavism.

Tamil's origin is independent of Sanskrit (which is from the Indo-European language family and the ancestor of many Indian languages), but it has borrowed a number of words from Sanskrit during recent centuries. The oldest Tamil text Tholkaappiyam is a grammar of Tamil.

2.1.1. Tamil Script

Indian scripts are generally written in non-cursive style, unlike the Latin alphabet, which is normally written in cursive style, rendering recognition difficult. However, Indian scripts pose a peculiar problem non-existent in European scripts, that is, the problem of composite characters. Unlike Latin alphabets where a single character represents a consonant or a vowel, in Indian scripts, a composite character represents either a complete syllable, or the coda of one syllable and the onset of another. Therefore, although the basic units that form composite characters of a

script are not that many ($O(10^2)$), these units by various combinations lead to a large number ($O(10^4)$) of composite characters.

The Tamil script has twelve vowels, eighteen consonants and one character, the āytam, which is classified in Tamil grammar as being neither a consonant nor a vowel, though often considered as part of the vowel set. The script is syllabic and not alphabetic (Iravatham, 2003). The complete script consists of the thirty-one letters in their independent form (Table 2.1) and an additional 216 combinatory letters (Figure 2.1) representing a total 247 combinations of a consonant and a vowel, a mute consonant or a vowel alone. These combinant letters are formed by adding a vowel marker to the consonant. Some vowels require the basic shape of the consonant to be altered in a way that is specific to that vowel. Others are written by adding a vowel-specific suffix to the consonant, yet others a prefix and finally some vowels require adding both a prefix and a suffix to the consonant. In every case, the vowel marker is different from the standalone character for the vowel.

Traditionally, a Tamil word is divided into a maximum of six parts, namely *pakuthy* (prime-stem), *sandhi* (junction), *vihaaram* (variation), *iTainilai* (middle part), *saariyai* (enunciater) and *vikuti* (terminator) in that order. For example, a word, *ndaTantanan (நடந்தான்,)* meaning '(He) walked', is made up of the morphemes நட+ந்(த்)+ஆன். The middle part and terminator are grammatical additions to the prime-stem. The middle part marks the tense, and the terminator marks the gender. Usually, the prime-stem is the main part of the word responsible for its meaning (Robin, 2012).

TABLE 2.1

TAMIL SCRIPTS

Type	No. of Scripts	Scripts
Vowels and Ayutha Letter	12+1	அ ஆ இ ஈ உ ஊ எ ஏ ஐ ஒ ஓ ஔ ஃ
Consonants	18	க் ங் ச் ஞ் ட் ண் த் ந் ப் ம் ய் ர் ல் வ் ழ் ள் ற் ன்
Consonants from Sanskrit	6	ஜ் ஶ் ஷ் ஸ் ஹ் க்ஷ்

Vowels → Consonants ↓	அ	ஆ	இ	ஈ	உ	ஊ	எ	ஏ	ஐ	ஒ	ஓ	ஔ
க்	க	கா	கி	கீ	கு	கூ	கெ	கே	கை	கொ	கோ	கௌ
ங்	ங	ஙா	ஙி	ஙீ	ஙு	ஙூ	ஙெ	ஙே	ஙை	ஙொ	ஙோ	ஙௌ
ச்	ச	சா	சி	சீ	சு	சூ	செ	சே	சை	சொ	சோ	சௌ
ஞ்	ஞ	ஞா	ஞி	ஞீ	ஞு	ஞூ	ஞெ	ஞே	ஞை	ஞொ	ஞோ	ஞௌ
ட்	ட	டா	டி	டீ	டு	டூ	டெ	டே	டை	டொ	டோ	டௌ
ண்	ண	ணா	ணி	ணீ	ணு	ணூ	ணெ	ணே	ணை	ணொ	ணோ	ணௌ
த்	த	தா	தி	தீ	து	தூ	தெ	தே	தை	தொ	தோ	தௌ
ந்	ந	நா	நி	நீ	நு	நூ	நெ	நே	நை	நொ	நோ	நௌ
ப்	ப	பா	பி	பீ	பு	பூ	பெ	பே	பை	பொ	போ	பௌ
ம்	ம	மா	மி	மீ	மு	மூ	மெ	மே	மை	மொ	மோ	மௌ
ய்	ய	யா	யி	யீ	யு	யூ	யெ	யே	யை	யொ	யோ	யௌ
ர்	ர	ரா	ரி	ரீ	ரு	ரூ	ரெ	ரே	ரை	ரொ	ரோ	ரௌ
ல்	ல	லா	லி	லீ	லு	லூ	லெ	லே	லை	லொ	லோ	லௌ
வ்	வ	வா	வி	வீ	வு	வூ	வெ	வே	வை	வொ	வோ	வௌ
ழ்	ழ	ழா	ழி	ழீ	ழு	ழூ	ழெ	ழே	ழை	ழொ	ழோ	ழௌ
ள்	ள	ளா	ளி	ளீ	ளு	ளூ	ளெ	ளே	ளை	ளொ	ளோ	ளௌ
ற்	ற	றா	றி	றீ	று	றூ	றெ	றே	றை	றொ	றோ	றௌ
ன்	ன	னா	னி	னீ	னு	னூ	னெ	னே	னை	னொ	னோ	னௌ

Figure 2.1: Combinant Tamil Letters

There are four kinds of words in Tamil. Among them, the root words or urichol were used only in ancient poems, and are not popular now. Thus, if it is excluded, then there exist only three types of words, namely, nouns, verbs and itaichol or particles (Varadarajan, 1988; Kalyanasundaram, 2015). The nouns indicate animate and inanimate categories of things, gender, number and person. tiNai is classified into uyartiNai (nouns denoting personal class of beings, including men, gods and demons) and akRiNai (inferior class of beings whether animate or inanimate or neuter). Higher categories of animate beings like human beings fall under uyrtiNai. Others, both animate and inanimate come under the category of akriNai. There are three genders in uyartiNai: masculine, feminine and neuter. Palar paal or neuter plural gender indicates many in number. Masculine and feminine genders in Tamil indicate only singular number. AkriNai is classified into onRan paal (singular of the impersonal class) and palavin pal

(plural of the impersonal class). Numbers are classified into one and many. Unlike Sanskrit, there is no dual number in Tamil.

There are three 'persons' in Tamil, namely, first person, second person and third person. Case inflexions are many in Tamil and their indicators form as suffixes in words. Distinction between animate and inanimate things and masculine and feminine genders are usually made according to the meaning of words. Verbs are classified into finite and infinite verbs. Most of the finite verbs are formed with suffixes, which indicate this animate or inanimate quality, as also gender. The gender is not distinguished both in abstract nouns and in relative participles. Both verbs and nouns are formed from verbal roots. But very few verbs are formed from noun roots. Particles have no meaning of their own, but acquire meaning when added to other words and help to differentiate their meanings too. Even meaningless words are regarded as particles.

Most of the words in Tamil are agglutinative in character, that is, case indicators, time and gender markers are affixed to root words. As a result, the formation of words becomes clear. Even the words in the classical literature are agglutinative in character. There is no distinction between the roots that were. used in ancient classics and those which are now in vogue. The root word which was used to mean 'food' in ancient classics was una. The one used in medieval period was either uN or uNTi, whereas the modern word for food is uNavu. In all these words, whether ancient or mediaeval or modern, the root word un is clear. Only the suffixes differ. Therefore, the Tamil of ancient poetry too begins to seem familiar after a while, if one reads the ancient classical poetry for a time. This is the reason why the Tamils of this century face little difficulty in understanding the sangam classics. It also accounts for the continuity that exists in Tamil literary growth. One finds it used in the poems of the hymnodists and Kamban, composed in the seventh century and the twelfth century respectively.

There is little difference in syntax between ancient Tamil and modern Tamil. Although over a period of time word forms have changed, the formation of syntax remains intact in all the Dravidian languages. An analysis on the continuous growth of Tamil language reveals the perceivable truth that there is little change in the formation of syntax both in the classical Tamil and the modern Tamil used today. Quite significantly for its age, Tamil seems to have undergone

minimal changes and adaptations over the years. Classical Tamil is quite comprehensible to speakers of the modern language.

2.2 SPEECH RECOGNITION

Speech recognition is the translation of spoken words into readable format, usually in American Standard Code for Information Interchange text format, which can be the final output or input to natural language processing. Automatic recognition of human speech by computers has been a topic of research for more than forty years.

Speech recognition is a very difficult task to be performed by a computer system. This situation is due to the variability in the way people speak, which results in complex speech signals that have to be processed by automatic speech/voice signals. They have to be processed by Automatic Speech Recognition System (ASRS) (Avci and Akpolat, 2006). In speech recognition area, there are several key areas of research for the current development of spoken language systems (Siafarikas *et al.,* 2004), and this section presents some of them.

2.2.1. Historical Background of Speech Recognition

Speech recognition systems have been researched and developed as early as the 1950s. Many of the attempts have achieved only limited success. One of the earliest mildly successful attempts was conducted by Biddulph and Balashek (1952), working at Bell Laboratories in 1952. Their ASR system could recognize the numbers 0 - 9 when spoken over a telephone.

There were many more developments through the 1950s and 1960s that led to the first usable ASR system that could recognize isolated words. These systems required extensive amounts of training and had a limited vocabulary (Rex and Thomas, 1978).

Over the past few decades, there have been important changes in the way the problem has been approached. They are briefly summarized below (http://www.nexus.carleton.ca/~kekoura/history.html).

- **The acoustic approach (pre-1960)**

The patterns of formant movements were analyzed in an attempt to recognize a word from a limited, predefined vocabulary (For example, digits between 1 and 10). The systems performed well, but only when used by the speaker they were designed for. The usefulness of this method was limited by the fact that acoustic patterns of a word spoken on different occasions differ in duration and intensity and the same word produced by different persons produces patterns differing in frequency content as well.

- **The pattern - recognition approach (1960-1968)**

Attempts were made to normalize the speech waveform in some way, so that comparisons with pre-defined patterns (words) could be made for a range of speakers. In particular, it was noted that the fundamental frequency could be used to normalize formant frequencies. Also, ways of normalizing the duration of patterns were investigated. The problem was still that such systems were only adequate for limited vocabularies.

- **The linguistic approach (1969 - 1976)**

The fact that, when two people communicate using speech, they must both use the same language was neglected by early recognizers. There are many sources of linguistic knowledge which could be used to enhance various systems, such as pre-stored dictionaries and the varying probabilities of a particular phoneme or word occurring after another one. This is referred to as phonotactics. Phonotactics deals with the rules that govern the combinations of the different phones in a language. There is a wide variance in such rules a Cross-Language. For example, the phone clusters /sr/ and /schp/ are common in Tamil and German respectively, but do not exist in English.

- **The pragmatic approach (1977-1980's)**

The vision for adapting speech-recognition technology existed long before any real-life practical adaptations were possible. Finally, during late 1980s and early 1990s, speech-recognition technology found its first niche in the marketplace. This niche comprised activities, in which users needed to operate computers, but did not have a free hand to punch keys or manipulate a mouse (Robert, 1995; Stephanie, 1997). An example from the music industry involves a keyboard player in the recording studio attempting to control other components,

particularly rhythm machines, without interrupting the play on the synthesizer. By rigging up an early speech-recognition system to the drum machine input, the musician was able to change tempos and rhythms by shouting speech commands, which were not picked up by the recording system (Ben and Nelson, 1999).

Early systems required users to learn how to "talk" to the computer rather than the computer learning to "listen" to the user. Users were forced to adapt to the technology by changing their way of speaking (Chris, 1996). Users with accents or even head colds found the systems to be frustratingly inaccurate. Compounding the problem were the characteristically slow processing times of the 486 computers of this era and the common inability of these speech-recognition systems to distinguish between background noise and the user's speech. Also, the costs of the early systems were excessive.

Continuous improvement in the technology of speech-recognition systems became imperative for hospitals, so that users would come to believe in the value of these systems. Therefore, the vocabularies built into these systems grew tremendously in both size and the degree to which they were tailored to the jargon and terminology of the medical profession (Lodato, 2002). The systems gradually became better at adapting to a particular user's speech, regardless of timbre, speech character, accents or head colds. Accuracy rates rose dramatically and doctors were no longer struggling for the "right" words for the system to understand and record. Newer systems even provided each user with an opportunity to "teach" the computer to understand their own speech.

Recent advances in speech recognition technology (You, 2009; Li *et al*. 2014) are creating a dynamic environment, as they are simple to use and use a hands-free approach to computing tasks. As the merger of large vocabularies and continuous recognition continues, more and more industries are moving towards speech recognition and the industry takes its place as a leader in the technology sector.

2.2.2. License Free Speech Recognition Software

The section describes some of the license free software available for speech recognition.

CVoiceControl (which stands for Console Voice Control) started its life as KVoiceControl (KDE Voice Control). CVoiceControl is an excellent starting point for experienced users looking to get started in ASR. It is not very user friendly, but once it has been trained correctly, it can be very helpful. (www.kiecza.de/daniel/linux/index.html).

GVoice is a speech ASR library that uses IBM's ViaVoice (free) Software Development Kit (SDK) to control Gtk/GNOME applications. It includes libraries for initialization, recognition engine, vocabulary manipulation and panel control. This software is primarily for developers (www.cse.ogi.edu/~omega/ gnome/gvoice).

The Institute for Signal and Information Processing at Mississippi State University has made its speech recognition engine available. The toolkit includes a front end, a decoder and a training module. It is a functional toolkit. The toolkit is available at www.isip.msstate.edu/project/speech.

Sphinx originally started at CMU and has recently been released as an open source. This is a fairly large program that includes a lot of tools and information. It includes trainers, recognizers, acoustic models, language models and some limited documentation. (www.speech.cs.cmu.edu/sphinx/Sphinx.html).

The Neural Inference COmputation (NICO) Artificial Neural Network toolkit is a flexible back propagation neural network toolkit optimized for speech recognition applications. (www.speech.kth.se/NICO/index.html)

2.2.3. Commercial Speech Recognition Software

Description of some of the popular commercial speech recognition software marketed in the industry is given below:

IBM support with Linux is made through a series of ViaVoice products. Their commercial product, IBM ViaVoice Dictation for Linux (www.4.ibm.com/software/speech/linux/dictation.html) performs very well, but has some sizeable system requirements compared to the more basic ASR systems (64M RAM and

233MHz Pentium). It also allows multiple users, and the package includes trainer, dictation system and installation scripts.

Babel Technologies has a Linux SDK available called Babear. It is a speaker independent system based on Hybrid Markov Models and Artificial Neural Networks technology. They also have a variety of products for Text to speech, speaker verification and phoneme analysis (www.babeltech.com).

Nuance is speech recognition software which offers a speech recognition/natural language product (currently Nuance 8.0) for a variety of linux platforms. It can handle very large vocabularies, and uses a unique distributed architecture for scalability and fault tolerance (www.nuance.com).

Abbot is a very large vocabulary, speaker independent ASR system. It was originally developed by the Connectionist Speech Group at Cambridge University. It was transferred (commercialized) to SoftSound. AbbotDemo is a demonstration package of Abbot. This demo system has a vocabulary of about 5000 words and uses the connectionist/HMM continuous speech algorithm (www.softsound.com).

Recent commercial examples also include Dragon Professional 13 (www.nuance.com/for-business/by-product/dragon/dragon-for-the-pc/dragon-professional), VoiceFinger (http://voicefinger.cozendey.com), ViaTalk (http://www.amazon.com/gp/product/B00I3JF87U), e-speaking (http://www.e-speaking.com) and tazti software (http://www.tazti.com/index.php).

2.2.4. Speech Recognition for Indian Languages

According to Kurian (2014), bridging the digital division between English speaking and non-English speaking people is vital, and which, to a certain extent, can be brought forward through the use non-English speech recognition software. This process is considered as more challenging with Indian languages where there are more than 1670 dialects of spoken form. This section presents some of the major works in automatic speech recognition used for Indian languages.

Samudravijaya *et al.* (2000) proposed a speaker independent, continuous speech recognition system for Hindi for recognizing spoken queries in Hindi in the context of railway reservation enquiry task. Neural network approach was proposed by Hassan *et al.* (2003) for Bengali phoneme recognition. In 2003, Saiprasad and Girija used Neural Networks for speech Recognition of Isolated Telugu Vowels.

Hegde *et al.* (2004) used joint features derived from the modified group delay function and MFCC for Continuous speech recognition in Telugu Language. Later, the same authors used a modified group delay function for continuous speech recognition in Telugu language. Rajput *et al.* (2004) developed a large-vocabulary continuous speech recognition system for Hindi. This system was trained on 40 hours of audio data and has a trigram language model trained with 3 million words and has a vocabulary size of 65000 words.

Balamurugan and Balaji (2006) proposed an online system for isolated word recognition on a vocabulary of 10 digits (0-9). This system acquired speech dynamically through microphones, and was implemented on a programmable chip. In the same year, a Bengali speech recognizer was built by training the HTK (Hidden markov model ToolKit) that can recognize any word in the dictionary (Hoque, 2006). After acoustic analysis of speech signal, the words were recognized.

Syama and Idikkula (2008) presented an isolated word and speaker independent speech recognition system for Malayalam language. Krishnan *et al.* (2008) developed a small vocabulary (5 words) speech recognition, using 4 types of wavelet for feature extraction and Artificial Neural Network (ANN) technique for classification and recognition of Malayalam words.

Banerjee *et al.* (2008) studied the effect of triphone based acoustic modeling over monophone based acoustic models in the context of continuous speech recognition in Bengali. Triphone clusters have been generated using decision tree based techniques. These triphone clusters have then been used to generate tied-state triphone based acoustic models to be used in a continuous speech recognizer.

Raza *et al.* (2009) developed a HMM-based large vocabulary automatic spontaneous Urdu speech recognition system with the help of Sphinx 3 trainer and decoder. Kalyani and Sunitha (2009) proposed a dictation system in Telugu language, using phoneme as a basic unit of recognition. The same authors, later in 2012 (Sunitha and Kalyani, 2012), proposed another system that uses syllable as the basic unit for Telugu language.

Sarfraz *et al.* (2010) proposed Large Vocabulary Continuous Speech Recognition (LVCSR) for Urdu language speech recognition using CMU Sphinx Open Source Toolkit. The system used a corpus of training data recorded in noisy environment so as to improve the robustness of speech recognition. Anusuya and Katti (2010) proposed a speech recognition scheme for isolated words in Kannada Language. This scheme was based on the Discrete Wavelet Transform (DWT) and Principal component Analysis (PCA).

Mathur *et al.* (2010) used Julius software tool for developing a domain specific speaker independent continuous speech recognizer for Hindi. Kumar (2010) compared the performance of Dynamic Time Wrapping (DTW) -based speech recognition and HMM-based speech recognition for Punjabi isolated words. Performance was in favor of DTW based recognizer.

Mandal *et al.* (2010) introduced the SPHINX3-based Bengali Automatic Speech Recognition system, which is the base for Shruti-II and E-mail applications. This system converted standard Bengali continuous speech to Bengali Unicode. Sukumar *et al.* (2010) presented recognition of the isolated question words from Malayalam speech query using DWT and ANN. A small vocabulary speech recognizer has been developed by Mohamed and Nair (2010), using Hidden Markov Models and Mel Frequency Cepstral Coefficients (MFCC), for recognizing words in Malayalam language.

Sarma and Sarma (2010) developed a numeric speech recognition system for Assamese language, which took into account the gender and mood variations during the recording of speech signals. The same authors (Sarma and Sarma, 2011) later proposed an optimal feature extraction algorithm and ANN based architecture for speech recognition in Assamese language. Mohanty and Swain (2010, 2011) developed an Oriya speech recognition system for visually impaired students.

Kumar and Aggarwal (2011) have developed a Hindi isolated word speech recognizer, using HTK on Linux platform, to recognize isolated words using acoustic word model. The system used a small vocabulary with 30 words and the HMM was train using 39 features combining MFCC, log energy and 1st and 2nd Order derivatives.

Aggarwal and Dave (2011) proposed statistical pattern classification to improve the speed of speech recognition in Hindi. The system reduced the time consumed in the likelihood evaluation of feature vectors, with the help of optimal number of Gaussian mixture components. They applied extended MFCC procedure by extracting 52 MFCC features (39 MFCC + 13 triple delta features), and then reduced them to 39 by using Hetroscedastic Linear Discriminant Analysis (HLDA) technique.

Mishra *et al.* (2011) developed a connected Hindi digits recognition system using robust feature extraction techniques and HTK recognition engine. Sivaraman and Samudravijaya (2011) compensated the mismatch between training and testing conditions with the help of unsupervised speaker adaptation to improve speech recognition. Maximum Likelihood Linear Regression (MLLR), a speaker adaptation technique requiring small amount of data, was used in this system.

Gawali *et al.* (2011) developed a system for recognizing isolated words of Marathi language, using Computer Speech Laboratory (CSL) for collecting speech data. A technique for fast bootstrapping of initial phone models of a Gujarati language was presented by Patel (2011). The training data for the Gujarati language was aligned, using an existing speech recognition engine for English language. This aligned data was then used to obtain the initial acoustic models for the phones of the Gujarati language.

Deka *et al.* (2011) proposed a Speech Recognition system using LPCC (Linear Predictive Cepstral Co-efficient) and MLP (Multilayer Perceptron) based Artificial Neural Network with respect to Assamese and Bodo language.

Hegde *et al.* (2012) developed an Isolated Word Recognition (IWR) system for the identification of spoken words in Kannada. Anusuya and Katti (2012) used statistical method to remove the silence from the speech signal. This method then used vector quantization technique

to identify the minimum speech patterns to generate training sample sets. The recognition was performed using statistical methods and clustering.

Shivakumar (2012) presented low cost, isolated word recognition of Kannada digits, for literate deaf people. The system used a feature vector, combining wavelets, MFCC, followed by Vector Quantization. Euclidean Distance measure is used to correlate the test speech signal with pre-recorded speech signals from the speech database. The nearest match is identified and its respective text equivalent is displayed.

VijaiBhaskar *et al.* (2012) also used the HTK toolkit for building a speech recognizer for Telugu language. Usharani and Girija (2012) proposed a system that used a modified static pronunciation dictionary for recognizing Telugu words. The modified dictionary was used in the decoder component of the speech recognition system and reduced the number of the confusion pairs. This improved the speed of the proposed system. Hegde *et al.* (2013) developed an isolated word recognizer for identification of spoken words in Kannada language using SVM classifier trained using MFCC features.

Speech recognition of Gujarati Language using neural network was presented by Patel Pravin and Jethva (2013). Bhattacharjee (2013) proposed two systems for recognizing the tonal words in Bodo language. The first method proposed a feature-level solution, while the second level proposed a model-level solution. The paper compared the relative merits and demerits of both the proposed methods.

Sunny *et al.* (2012, 2013) presented a speech recognition system for isolated word recognition in Malayalam. This system studied the effect of wavelet-based features, LPCC and used artificial neural network for recognition.

2.2.5. Tamil Speech Recognition System

Nayeemulla and Yegnanarayan (2001) proposed a speech recognition system for some specific domains of Tamil language. Sararwathi and Geetha (2004) developed a language model for Tamil speech recognition system. Madhuresh *et al.* (2003) discussed the various pros and cons of information technology systems developed for Indian languages.

Lakshmi *et al.* (2006) presented a technique for building a syllable based continuous speech recognizer for un-annotated transcribed Tamil speech data. The system used two segmentation algorithms, namely, group-delay based two-level segmentation algorithm and rule-based text segmentation algorithm. The first algorithm was used to extract the syllable units from the speech data, while the second algorithm was used to automatically annotate the text corresponding to the speech into syllable units. Isolated style syllable models were built using Multiple Frame Size (MFS) and Multiple Frame Rate (MFR) for all unique syllables by collecting examples from annotated speech.

Chandrasekar and Ponnavaikko (2008) presented a continuous speech recognition system in Tamil. This system first segmented the speech signal into words and then to characters. The result was then used by Back Propagation Neural Network to recognize the individual characters. Thangarajan *et al.* (2008) developed a small vocabulary word-based and medium vocabulary triphone based continuous speech recognizer for Tamil language. The same authors (Thangarajan *et al.,* 2009) later extended this work for continuous speech recognition in Tamil language, using syllable as a sub-word unit for building acoustic model. Initially, a small vocabulary context independent word model and a medium vocabulary context dependent phone model were developed. Then, an algorithm based on prosodic syllable was proposed for recognition.

According to Scharenborg (2007) and Schutte (2009), the accuracy of today's speech recognition system still continues to lag behind human performance and there is still a considerable gap between human and machine performance. This is more prominent with Tamil speech recognition, where the research is still more active.

Dharun and Karnan (2012) developed a system for Tamil language word and numeral recognition using MFCC and DTW. Vimala and Krishnaveni (2012) presented a Continuous Speech Recognition (CSR) system for Tamil language using Hidden Markov Model (HMM) approach that used MFCC features. Vimala and Radha (2012) proposed a speaker independent isolated speech recognition system for Tamil language.

According to Karpagavalli et al. (2012a), DTW-based Speech Recognition System improves the accuracy of isolated Tamil Digits speech recognition. Karpagavalli et al. (2012b) developed another system for speaker independent and isolated Tamil digit recognition using

template based and HMM based approaches. Gnanathesigar (2012) proposed a Tamil speech recognition system using semi-continuous models based on Hidden Markov model.

An isolated Tamil word recognizer was built using the HTK tool (Akila and Chandra, 2013). This recognizer was trained with the grain names in Tamil spoken by 2 female speakers. In the following year, Pushpa *et al.* (2014) proposed a Tamil speech processing and recognition system based on Back Propagation Neural Network (Multilayer feed forward classifier) with semi-supervised training. The system first used four filters, namely, pre-emphasis, median, average and Butterworth band stop to remove background noise. Then, the LPCC features were extracted to train the classifier. The system used a technique called training deep neural networks as data driven feature front end for large vocabulary continuous speech recognition.

Even though many speech processing tasks, like speech and word recognition, have reached satisfactory performance levels on specific applications, and although a variety of commercial products have been launched in the last decade, many problems are yet to be solved in speech recognition. Hence, it is an open research area where absolute solutions have not been found yet.

2.3. INFORMATION RETRIEVAL (IR) APPROACHES

Information retrieval, a word coined first by Calvin (1950), has become a mature technology to discover relevance among retrieved information from different sources, not only in the news domain but also in special domains (Mavaluru, 2014). This section presents general IR approaches along with the key techniques used by IR models. In general, there are four frequently used IR models (Yang *et al.*, 2014). They are, Boolean model, vector space model, probabilistic models and language models.

2.3.1. Boolean Model

The Boolean retrieval model is a model for information retrieval where queries are presented in a Boolean expression of terms. The Boolean operators include AND, OR and NOT, which connect terms to form a query. The operators AND and OR affect performance in opposite ways. The more OR operators in a query, the more extraneous items are retrieved, which reduces the retrieval precision. On the other hand, the AND operator tends to increase retrieval precision,

while recall declines. The advantage of the Boolean model is the high precision for high recall searches. This model has the following issues:

- Boolean queries are difficult to formulate. Ngai *et al.* (2007) illustrated several operations needed to formulate a Boolean query including removal of high-frequency terms, additional synonyms and alternate spellings. Moreover, it is hard to insert extra terms that are not originally included.
- Most applications of the Boolean model do not provide the assignment of term weights, on which the query-document relevance measurement depends. Gatian (1994) extended the base Boolean model to add term weighting and output ranking features.
- The retrieved documents are usually presented in a random order, that is, with no ranking, because the Boolean model does not provide an estimate of the query-document relevance.
- The size of the subset of documents to be returned is difficult to control.
- It is difficult or impossible to find a satisfactory middle ground between AND and OR. Madhavi *et al.* (2005) proposed a compromise by the use of a query formulation that is neither too broad nor too narrow.

2.3.2. The Vector Space Model

The Vector Space Model (VSM) (Kembellec *et al.*, 2009) uses a ranking algorithm that tries to rank information according to the overlap between the query terms and information. In this model, all queries and information are represented as vectors in |V|-dimensional space, where V is the set of all distinct terms in the document. The vector space model requires the following calculations, where the model for term weight is called Term Frequency/Inverse Document Frequency (TF/IDF) model. Compared with the Boolean retrieval model, the vector space model has the advantages listed below:

- It is a simple model based on linear algebra
- Term weights are not binary
- It provides for computing a continuous degree of similarity between queries and documents

- Ranking documents is performed according to the similarity measure
- It is possible to only match a part of a document.

However, there are a few limitations to the vector space model as listed below.

- Terms are assumed to be independent of each other
- Long documents are poorly represented due to poor similarity values
- Query terms must precisely match document terms, otherwise substrings of terms could result in a false match
- It is difficult to take into account the order of terms appearing in a document.

The vector space model is applied not only to document or text retrieval but also to other information retrieval related applications, such as topic tracking (Gluck, 1996; Ma *et al.,* 2014), text categorization (Goenka *et al.,* 2010; Trejo *et al.,* 2015), web page classification (Gawande and Suryawanshi, 2015) and collaborative filtering (Wang *et al.,* 2012).

2.3.3. Probabilistic Retrieval Model

Probabilistic retrieval models are used to estimate the probability of documents being relevant to a query (Xinkai, 2011). This probability is then used to rank all documents in response to a query. This model is also referred to as the probability ranking principle.

This model assumed that the terms were distributed differently in relevant and non-relevant documents. Probabilistic models are based on the "Probability Ranking Principle" (PRP) (van Rijsbergen, 1979; Zuccon *et al.,* 2009), which proposed that all documents can be simply ranked in decreasing order of the probability of relevance with respect to the information need. Ripley (2008) proved that the PRP is optimal, but it requires that all probabilities are known correctly, which is in practice impossible.

In order to estimate the probability of relevance of the document to a query, the Binary Independence Model (BIM) (*https://en.wikipedia.org/wiki/ Binary_Independence_Model)* was introduced. "Binary" means that documents and queries are both represented as binary term vectors. "Independence" indicates the terms occurring in documents independently, that is the presence or absence of a term in a document, are independent of the presence or absence of any

other term. The problem of BIM is that it was originally designed for short catalogue records and abstracts of fairly consistent length, and does not consider the term frequency and document length carefully.

The 2-Poisson model proposed by Bookstein and Swanson (2007) showed that a term plays two different roles in documents. The first role is in documents with a low average number of term occurrences, the term should not be used as an index term and the second role is in documents with a high average number of term occurrences, the term is a good index term.

Robertson and Walker (1994) presented an IR model approximating the 2-Poisson model, known as the Okapi weighting scheme. Among the Okapi BM series, Okapi BM25 (Robertson *et al.,* 1992) is the optimal result. Its name is derived from "BM", which means "Best Match" and a version number of the last trial, 25, which was a combination of BM11 and BM15 (Robertson *et al.,* 1994).

Logistic Regression (Cooper *et al.,* 1994) and Pircs (Kwok, 1996) are two well known probabilistic models. Although they perform well, studies like the one by Luk and Kwok (2002), show that the 2-Poisson model with Okapi BM25 weighting scheme exceeds other probabilistic models.

The probabilistic information retrieval model has been enhanced and used to improve the process of information retrieval in many manners. Xu and Akella (2010) improved probabilistic information retreival by modeling burstiness of words using Dirichlet Compound Multinomial (DCM) distribution based on the Polya Urn Scheme and a pseudo-relevance feedback algorithm based on the mxiture modeling of DCM. He *et al.* (2011) used proximity among query terms to improve retrieval performance. For this purpose, an improved version of the classical BM25 model was proposed. This method used several methods including a window-based N-gram Counting method, Survival Analysis over different statistics, including the Poisson process, an exponential distribution and an empirical function.

Maitah *et al.* (2013), on the other hand, proposed a novel algorithm that used adaptive genetic algorithm for improving the effectiveness of information retrieval. Zhao *et al.* (2014) used this model by introducing a new concept cross term, to model term proximity, with the aim

of boosting retrieval performance. Marin *et al.* (2014) improved Deviation from Randomness model in cognitive contexts by adjusting the documents length normalization. Deviation from randomness (DFR) is a methodology for modeling unstructured information retrieval using a weighting to generate a relevance ranking of documents based on the concepts of information content and information gain.

2.3.4. The Language Model

The language model is based on the idea that a document is a good match to a query if the document model is likely to generate the query, which will in turn happen if the document contains the query words a number of times (http://nlp.stanford.edu/IR-book/html/htmledition/language-models-for-information-retrieval-1.html). In practice, the language model for IR is based on the unigram model, because the unigram model is sufficient to judge the topic of a text. In addition, the unigram model is more efficient to estimate and apply than higher-order models.

The language model has many variant realizations. Lafferty and Zhai (2001) proposed three ways to establish language models. They are the query likelihood language model that uses documents to generate a query, document likelihood language model, where the query model is used to estimate documents and the model comparison approach.

In the query likelihood model, a language model Md, constructed from each document d in the collection, is applied to model the query generation process. The probability P (d|q), where the probability of a document is interpreted as the likelihood that it is relevant to the query, is used to rank relevant documents.

An alternative language model is the document likelihood model. The problem of this approach is that there is much less text available to estimate a language model based on the query text. Queries are in common very short. For example, Jansen *et al.* (2005) reported that 20% of web queries in 2002 contained only a single term. The sparseness of query texts causes the models derived from queries to be unreliable. Lavrenko (2004) reported that document likelihood models perform poorly.

The third approach to the language model is the model comparison. Lafferty and Zhai (2001) used the Kullback-Leibler (KL) divergence between the document language model and the query likelihood model to model the risk of returning a document d as relevant to a query q. The work demonstrated that the model comparison approach outperforms both query likelihood models and document likelihood models. Manning and Schutze (1999) pointed out that KL divergence is not symmetric and does not satisfy the triangle inequality, and thus is not a metric. Therefore, the problem of using KL divergence as a ranking function is that scores are not comparable across queries. Kraaij and Spitters (2003) suggest that the similarity can be modeled as a normalised log-likelihood ratio.

All language models are faced with the zero-frequency problem, in which the frequency of a term is zero because the term does not occur in the document collection. Thus the probability involving this term is zero. Smoothing is a solution to this issue. In general, all smoothing techniques are attempting to discount the probabilities of the terms appearing in the documents and then to assign the extra probabilities to the unseen terms.

Considering the efficiency of computations over a large collection of documents, there are three smoothing techniques widely applied in language models, namely Jelinek-Mercer smoothing, Dirichlet smoothing and two-stage smoothing. A detailed review of these smoothing methods are provided by Chen and Goodman (1996) and Zhai and Lafferty (2004). An alternative smoothing technique is Dirichlet smoothing. After applying Dirichlet smoothing, the query likelihood language model can be computed. The two-stage smoothing method is an improvement of Dirichlet smoothing, which uses a two-stage strategy. In the first stage, a query language model is smoothed using Dirichlet smoothing; and in the second stage, it is smoothed using Jelinek-Mercer. It has the advantage over Dirichlet smoothing in that no tuning is needed and it performs well.

Several authors have used document language model for improving information retrieval. Tao *et al.* (2006) proposed a document expansion technique to deal with the problem of insufficient sampling of documenting. This method constructed a probabilistic neighborhood for each document and then expanded the document with its neighborhood information. The expanded document provided more accurate estimation of the document model, thus improving

the information retrieval process. Puurula (2013) proposed several improvement methods to improve the language models to information retrieval. The methods used include Pitman-Yor Process smoothing, TF-IDF feature weighting, model-based feedback and ranking. Cummins *et al.* (2015) propose a Polya Urn document multilingual language model using TREC collection. In this method, the document generation is modeled as a random process with reinforcement (a multivariate Polya process) and developed a Dirichlet compound multinomial language model to capture word burstiness directly.

2.3.5. Syntactic Models

All the above mentioned models neglect an important factor, called linguistic or syntactic characteristics of a language. This model parses texts using linguistic knowledge (Wang, 2011). Usage of linguistic and syntactic information with languages were initially used for indexing. Examples include the proposal of Baxendale (1958), Clarke and Wall (1965), Hillman (1968), Maeda *et al.* (1980), Salton (1983), Jones (1983), Dillon and McDonald (1983) and Piternick (1984).

However, researchers did not stop investigating the effect of linguistic information for indexing alone. They applied the same for information retrieval. Early works includes works of Fagan (1987), Smeaton *et al.* (1994), Evans and Zhai (1996), Pohlmann and Kraaij (1997). Liu *et al.* (2007) used a syntactic tree structured representation of documents and queries and matched documents and queries with tree comparison algorithms. On the other hand, Lioma and Ounis (2008) used a syntactic-based query reformulation technique for IR.

Straková and Pecina (2010) retrieved syntactic dependency information automatically from both documents and queries and used Czeh test collection for adhoc tracking. Ferrandez (2011) performed information retrieval, using lexical and syntactic knowledge, extracted using POS-tagger and syntactic chunker, and used similarity measures for estimating the dependency information between entities and terms. The syntactic information was retrieved using syntactically based query reformulation technique, which was compared using probabilistic pseudo-relevance feedback technique.

Ionescu *et al.* (2014) proposed three techniques for incorporating syntactic metadata for document retrieval. The first technique involves just a syntactic analysis of the query and it generates a different weight for each term of the query, depending on its grammar category in the query phrase. These weights will be used for each term in the retrieval process. The second technique involves a storage optimization of the system's inversed index, that is, the inverse index will store only terms that are subjects or predicates in the document they appear. Finally, the third technique builds a full syntactic index, meaning that for each term in the term collection, the inverse index stores besides the term-frequency and the inverse-document-frequency, also the grammar category of the term for each of its occurrences in a document.

A comparison of semantic and syntactic IR was performed by Gupta and Garg (2011). They concluded that syntactic retrieval system fluctuating recall and precision rate. Moreover, they reported that syntactic methods are heavily dependent on the quality of language analyzers and dictionaries.

2.4. CROSS-LANGUAGE INFORMATION RETRIEVAL

The methods discussed in previous section are generally known as monolingual IR, where the non-English documents are treated as unwanted noise (Abusalah *et al.*, 2005). As mentioned earlier, in CLIR, queries and documents are expressed in different languages. The CLIR uses the same indexing algorithms and retrieval models as classic IR but also employs various more sophisticated methods to improve retrieval performance. The basic idea and technique in performing cross-lingual information retrieval is translation (Ren and Bracewell, 2009), which is the task of translating query or document manually or automatically. However, translation is not the only approach to CLIR. This section presents the history of CLIR, followed by description of non-translation and translation based techniques.

2.4.1. History of CLIR

Although the majority of studies in IR concern monolingual IR, CLIR problems attracted research interests from as early as 1960s (Salton, 1970). Since then, a number of attempts have been made on CLIR and MLIR, in particular, in the area of library science. Readers can find a summary of the early attempts in this area in (Oard and Dorr, 1996).

Research in CLIR was predominant from mid-1990s when World-Wide Web started becoming popular. Documents in English and other languages became publicly available. Even though the majority of searches on the Web was (and still is) monolingual, there were needs for retrieving documents in other languages. Investigations became intensified from 1997 when CLIR experiments were officially conducted in Text Retrieval Conference (TREC)-6 (http://trec.nist.gov) organized by the National Institute of Standards and Technology (NIST) (Voorhees and Harman, 1997).

Even earlier in TREC-4 and TREC-5, while retrieval experiments on Spanish documents were conducted, some groups (Davis and Dunning, 1995) already carried out experiments on several ways to translate queries from English to Spanish. More CLIR experiments have been carried out between more European languages since TREC-7: English, French, German, Italian, Dutch and so on. Table 2.2 summarizes the monolingual IR on languages other than English and CLIR experiments in TREC.

CLIR experiments on European languages started in CLEF (Cross-Language Experiment Forum) in 2000 (http://www.clef-campaign.org). The first experiments dealt with English, German, French and Italian documents using queries in Dutch, English, French, German, Italian, Spanish, Swedish and Finnish. Then, more languages were added in the following years. Examples include languages like Spanish, Dutch, Swedish, Finish, Portuguese, Russian, Bulgarian and Hungarian. From 2005, multilingual retrieval has been conducted on a Web collection. EuroGov, collected from a number of government websites in Europe. In CLEF 2007, Indian languages are studied that included languages like Hindi, Telugu and Marathi.

The NTCIR (http://research.nii.ac.jp/ntcir) series of workshops started in 1999. They are organized by the National Institute for Informatics (NII) of Japan. They focus on Asian languages, in addition to English, Japanese, Chinese and Korean. New Asian languages are also being considered that included Vietnamese and Mongolian.

TABLE 2.2

CLIR EXPERIMENTS IN TREC

TREC	LANGUAGES AND DOCUMENT COLLECTIONS	QUERIES
TREC-3 (1994)	Spanish (monolingual): *El Norte* Newspaper	SP 1-25 (Spanish)
TREC-4 (1995)	Spanish (monolingual): *El Norte* Newspaper	SP 26-50 (Spanish)
TREC-5 (1996)	Spanish (monolingual): *El Norte* newspaper and *Agence France Presse* Chinese (monolingual): *Xinhua News agency, People's Daily*	SP 51-75 (Spanish) CH 1-28 (Chinese)
TREC-6 (1997)	Chinese (monolingual): The same documents as TREC-6 CLIR: English: *Associated Press* French, German: *Schweizerische Depeschenagentur (SDA)*	CH 29-54 (Chinese) CL 1-25 (English, French)
TREC-7 (1998)	CLIR: English, French, German, Italian: *Schweizerische Depeschenagentur (SDA)* + German: *New Zurich Newspaper (NZZ)*	CL 26-53 (Several languages)
TREC-8 (1999)	CLIR in English, French, German, Italian: The same document sets as in TREC-7	CL 54-81 (Several languages)
TREC-9 (2000)	English-Chinese: Chinese newswire articles from Hong Kong	CH 55-79 (English, Chinese)
TREC 2001	English-Arabic: Arabic newswire from *Agence France Presse*	1-25 (English, Arabic)
TREC 2002	English-Arabic: Arabic newswire from *Agence France Presse*	26-75 (English, Arabic)

In addition to the above experiments on CLIR, there are also initiatives to develop methods for IR in different languages. For example, a series of experiments on Chinese Web IR have been organized by Peking University (http://net.pku.edu.cn/~webg/cwt) since 2004. Forum for Information Retrieval Evaluation (http://www.isical.ac.in/~clia) started in 2008, aiming at testing IR and CLIR techniques for Indian languages Hindi, Bangla, Marathi, Tamil, Telugu, Punjabi and Malayalam. All these experiments have triggered a tremendous amount of research

work on CLIR and Multi-Lingual Information Retrieval (MLIR), and contributed significantly to the development of new techniques for CLIR and MLIR.

If CLIR effectiveness (measured in terms of mean average precision-MAP) was much lower than that of monolingual IR at the beginning (around 50%), the difference between them has been much reduced. In the current state of the art for well-studied languages, CLIR's effectiveness is close to that of monolingual IR. This shows the maturity of CLIR techniques. Another sign of the maturity of technologies of CLIR and MLIR is the fact that commercial companies started to offer products for them. For example, Yahoo! started to offer multilingual search from 2006. It allows automatically translating queries in French and German to four other languages, namely, English, Spanish, Italian and French/German and to retrieve documents in these languages.

Google also started offering CLIR facilities for a number of languages from 2007. First, the user's query is translated into one of the target languages. The retrieved documents in the latter language are then translated back to the query language using an MT system. The quality of the translations by Yahoo! and Google is variable according to the topic areas and the language. However, these tools allow the users to access more easily the documents written in different languages and to get a quick idea of their contents. They provide the prerequisite for practical uses of CLIR. Thus, the major events in the history of CLIR can be summarized as below.

- Started in 1960s, CLIR research has been particular active in the area of library science.
- Took off from mid-1990s with the World-Wide Web.
- The year 1997 started CLIR at TREC organized by the National Institute of Standards and Technology (NIST)
- The year 2000 envisaged, CLEF (Cross-Language Experiment Forum) CLIR experiments on European languages. The first experiment focused on EN, GE, IT, SP, Swedish and Finish and later focused on many languages. In CLEF 2007 Indian languages were also considered.
- The year 1999 started CLIR with NTCIR that is focused on Asian languages.

- The FIRE started at 2008, with the aim of building south Asian languages and continuously met new challenges in Multi-lingual information needs.

2.4.2. Tamil-based CLIR Systems

Telugu-Tamil machine translation system was developed at CALTS, which used the Telugu morphological analyzer and Tamil generator. The backbone of the system is Telugu-Tamil dictionary (Murthy and Deshpande, 1998).

Germann (2001) reported a statistical-based system which included the creation of a small parallel Tamil-English corpus. Parallel corpus of about 100,000 words on the Tamil side was created, using several translators. In order to improve the text coverage, a simple text stemmer for Tamil, which based on the Tamil inflection tables, was designed.

Renganathan (2002) developed an English-Tamil web based system based on rules. This system contains around five thousand words in lexicon, and a wide range of transfer rules written in Prolog. This system also considers frequently occurring English structures mapped to corresponding Tamil structures. Two types of lexicons were used, one is based on grammatical categories of head and target words and the other is based on semantic and syntactic properties of words. The morphological transducer was built as part of this system using this information to generate correct inflectional forms. It is constructed following the concepts of the theory of lexical phonology, which accounts for the interrelationship between phonological and morphological rules in terms of lexical and post lexical rules.

Gey (2002) reported a translation system for Tamil language. The system used a corpus of Tamil news stories from Thinaboomi website having more than 3,000 news stories in the Tamil language and provided a rich source for modern Tamil linguistic studies and retrieval. This corpus has been used to develop an experimental statistical machine translation system from Tamil to English by the Information Sciences Institute (http://www.isi.edu), one of the leading machine translation research organizations.

Chellamuthu (2002) explained the role of Machine translation in information dissemination and a brief history of MT and its strategies. The various components and functions

of an early MT system developed in Tamil University, Tanjore for Russian to Tamil translation is also explained. The Russian to Tamil MT system uses an intermediate language with a syntax more related to Target Language. The Russian to Tamil system consists of various functional components such as a preprocessor, parser, lexical analyzer, bi-lingual dictionary, morphological analyzer, and translator and generation modules. Here, the primary task was to analyze the input text, parse the sentences, analyze the words lexically and morphologically, conceptualize the sentences, table look up using bi-lingual dictionary and translate the input word using the linguistic knowledge already defined in the system.

Weerasinghe (2004) developed a system for Sinhala to Tamil Language. In this method, corpora were utilized from newspaper of Sri Lanka published in both languages. The system also used a corpora collected from a website that contains translations of English articles into Sinhala and Tamil. The system used a sentence boundary detection algorithm using a semi-automatic approach and the sentences were aligned manually. The CMU-Cambridge Statistical Language Modeling Toolkit (version 2) was used to build n-gram language models.

A Telugu-Tamil translation system was proposed by Bandyopadhyay (2004) using Telugu morphological analyzer and Tamil generator. English to Hindi, Kannda and Tamil was developed using language-pair example based approach by Balajapally *et al.* (2006). It is based on a bilingual dictionary comprising of sentence-dictionary, phrases-dictionary, words-dictionary and phonetic-dictionary is used for the machine translation. Each of the above dictionaries contains parallel corpora of sentence, phrases and words, and phonetic mappings of words in their respective files. Example Based Machine Translation (EBMT) has a set of 75000 most commonly spoken sentences that are originally available in English. These sentences have been manually translated into three of the target Indian languages, namely Hindi, Kannada and Tamil.

Kumaran and Kellner (2007) proposed a framework for translation from English to Hindi, Tamil, Arabic, Japanese and backward translation from Hindi, Tamil, Arabic, Japanese to English. Afraz and Sobha (2008) used statistical technique combined with n-gram based approach for Tamil to English translation that can be used in CLIR. This algorithm used n-gram frequencies to find the probabilities and each pattern of consonant-vowel in the word.

Ganesan and Siva (2007) proposed a way of processing multilingual information wherein the backend uses English language and the front end uses local language like Tamil. For searching multilingual information, there exist two methodologies, one based on Phonemes and another based on semantic matching. For semantic matching query crawler algorithm was proposed and for phonemes word crawler was proposed.

Janarthanam *et al.* (2008) proposed an efficient algorithm for English named entities to Tamil. They used a compressed word format algorithm and rewrite rule for replacing characters in named entities.

A Tamil-Hindi translation system was developed by Sobha *et al.* (2009), based on Anusaaraka and used lexical-level translation along with morphological analyzer, Tamil-Hindi bilingual dictions. The system also had a prototype for English to Tamil translation using a small set of transfer rules. Thenmozhi and Aravindan (2009) presented a machine translation approach combined with ontology and bilingual dictionary for performing Tamil to English translation and retrieval.

Vijaya *et al.* (2010) developed English to Tamil system using a Rule based approach which used 48 decision tree classifiers for classification. The translation process consisted of four phases, namely, pre-processing, feature extraction, training and translation. Saraswathi *et al.* (2010) used a bilingual dictionary for machine translation and construction of ontological tree to perform Tamil to English translation and retrieval.

Loganathan R (2010) developed the English-Tamil machine translation system using rule-based and corpus-based approaches. For rule based approach, structural difference between English and Tamil is considered and syntax transfer based methodology is adopted for translation.

Saravanan *et al*. (2010) developed a Rule based Machine translation system for English to Tamil. Using statistical machine translation approach, Google developed a web based machine translation engine for English to Tamil language. This system is also having the facility to identify the source language automatically.

Computational Engineering and Networking research centre of Amrita School of Engineering, Coimbatore, proposed an English-Tamil translation memory system (Harshawardhan *et al.*, 2011). The system was based on phrase based approach by incorporating concept labeling using translation memory of parallel corpus. 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 an accuracy of 70% .

Rao and Sobha (2012) also used bilingual dictionary and ontology for the translation and retrieval of English news magazine when supplied with Tamil queries. Technology Development for Indian Languages (TDIL) recently launched Sandhan (http://pib.nic.in/newsite/ erelease.aspx?relid=87874).

Sandhan is a mission mode project under TDIL Programme. Its main objective is to develop a monolingual search system for tourism domain in five Indian languages viz., Bengali, Hindi, Marathi, Tamil and Telugu. The system has been developed to satisfy the user information need in tourism domain. Sandhan has the capability to process the query based on its language and retrieve results from the respective language. An additional UNL based semantic search facility has been provided for Tamil language. Many of the Indian language web pages are in custom fonts that make the system difficult for retrieving documents Sandhan uses a font transcoder that converts the custom fonts into Unicode fonts for processing (Sourabh, 2013).

2.5. CHAPTER SUMMARY

From the literature study, it can be understood that Tamil language, belonging to the Dravidian scripts of Southern India, is one of the classical languages of the world with a literary history of more than two millenniums spanning from the sangam age (300 BC – 200 AD). It is one of the oldest languages that has several million speakers across the world and is an official language in countries such as Sri Lanka, Malaysia, Singapore and Tamil Nadu State of India. The penetration of Information Technology (IT) becomes harder in a country such as India where the majorities read and write in their native language. Therefore, enabling interaction with computers in the native language is absolutely necessary, and this research work proposes algorithms to facilitate this.

Cross-lingual information retrieval for foreign languages like English, French, and Chinese, etc. has been an appealing area for researchers from long time. But Indian languages have grabbed attention only a decade back. The work done by researchers show mixed results in terms of improvement over monolingual retrieval in Indian language perspective. In particular, majority of the limited proposed approaches related to Tamil, involves translation from English query to Tamil, while only very few have focused on Tamil to English. Even minimal are system that combines Tamil speech recognition with Tamil to English CLIR. In this research, a speech query based Tamil-English CLIR system is proposed, and the various steps involved in the proposed system, along with the research methodology, and a brief introduction to the various techniques that are used in the proposed system are presented in the next chapter, Chapter 3, **Methodology**.