A MACHINE TRANSLATION SYSTEM FOR ENGLISH – TAMIL USING UNIVERSAL NETWORKING LANGUAGE(UNL)

by

KASHYAP K 2009103555
PAVITHRA S 2009103036

A project report submitted to the

FACULTY OF INFORMATION AND COMMUNICATION ENGINEERING

in partial fulfillment of the requirements for the award of the degree of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

ANNA UNIVERSITY, CHENNAI – 25

MAY 2013

BONAFIDE CERTIFICATE

Certified that this project report titled **A MACHINE TRANSLATION SYSTEM FOR ENGLISH – TAMIL USING UNIVERSAL NETWORKING LANGUAGE(UNL)** is the *bonafide* work of **KASHYAP K (2009103555)** and **PAVITHRA S (2009103036)** who carried out the project work under my supervision, for the fulfillment of the requirements for the award of the degree of Bachelor of Engineering in Computer Science and Engineering. Certified further that to the best of my knowledge, the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or an award was conferred on an earlier occasion on these or any other candidates.

Place: Chennai
Date:

Rajeswari Sridhar
Assistant Professor(Senior Grade)
Department of Computer Science and Engineering
Anna University, Chennai – 25

COUNTERSIGNED

Head of the Department,
Department of Computer Science and Engineering,
Anna University Chennai,
Chennai – 600025

ACKNOWLEDGEMENT

We express our deep gratitude to our guide, **Dr. Rajeswari Sridhar** for guiding us through every phase of the project. We appreciate her thoroughness, tolerance and ability to share her knowledge with us. We thank her for being easily approachable and quite thoughtful. Apart from adding her own input, she has encouraged us to think on our own and give form to our thoughts. We owe her for harnessing our potential and bringing out the best in us. Without her immense support through every step of the way, we could never have it to this extent.

We are extremely grateful to **Dr. C. Chellappan**, Head of the Department of Computer Science and Engineering, Anna University, Chennai – 25, for extending the facilities of the Department towards our project and for his unstinting support.

We express our thanks to the panel of reviewers **Dr. Arul Siromoney, Dr. Geetha Palanisamy and Dr. S. Sudha** for their valuable suggestions and critical reviews throughout the course of our project.

We thank our parents, family, and friends for bearing with us throughout the course of our project and for the opportunity they provided us in undergoing this course in such a prestigious institution.

Kashyap K **Pavithra S**

ABSTRACT

This document proposes an English - Tamil Machine Translation system using Universal Networking Language (UNL) as the intermediate. This approach is a hybrid of rule and knowledge-based approaches to Machine Translation (MT). The input English sentence is converted to UNL which is then converted to a Tamil sentence preserving the meaning of the input sentence.

UNL, Universal Networking Language, is a declarative formal language specifically designed to represent semantic data extracted from natural language text. The process of conversion from any natural language to UNL is called Enconversion and the reverse process is called Deconversion. Thus translation from one language to another (English to Tamil) can be accomplished by using an Enconverter of source language (English) and Deconverter of target language (Tamil).

Using this approach we achieved a BLEU score of 0.581 which denotes that most of the information in the input sentence is retained. The scores obtained using UNL based approach shows that it is the best approach to Machine Translation when compared with other existing approaches.

ABSTRACT

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

MT	Machine Translation
UNL	Universal Networking Language
UW	Universal Word
HW	Head Word
PBT	Phrase Based Machine Translation
EBMT	Example Based Machine Translation
RBMT	Rule Based Machine Translation
CBMT	Context Based Machine Translation
POS	Part Of Speech
JAWS	JAVA API for Wordnet Searching
PNG	Person Number Gender
KB	Knowledge Base
BLEU	BiLingual Evaluation Understudy
WER	Word Error Rate

CHAPTER 1

INTRODUCTION

1.1 PROBLEM DOMAIN

Natural Language Processing is a field of computer science which deals with interactions between the computer and human languages. It deals with making the computer to understand and interpret human language. Computational linguistics is the term used in Natural Language Processing. It is an interdisciplinary field dealing with the statistical or rule-based modelling of a natural language from a computational perspective. Machine translation is a sub-field of computational linguistics that investigates the use of software to translate text or speech from one natural language to another.

Translation from one language to another is usually done using a dictionary (source to target language) which is a tedious job. Even though we try to do a word by word translation, translation at the sentence level is essential to convey the information in its fullest form by taking into consideration the context at which every word is used. As the process of sentence level translation is tedious for human beings, this process can be automated. For a machine to do translation, in addition to the dictionary, we need to provide the grammar of source and target languages to do sentence level translation. Word sense disambiguation, preserving the meaning of the sentence, grammatical correctness of the translated sentence, handling various dialogue acts and handling the differences between the source and target languages are some of the major

issues in machine translation. Researchers have been working on automated translation between languages for nearly two decades. Many approaches have been described for Machine Translation. A few of them are Phrase Based MT, Context Based MT, Rule Based MT, Example Based MT and Hybrid approaches.

In this project, we have developed a Machine Translation system that would translate an English sentence to Tamil using Universal Networking Language (UNL) as an intermediate. English is the most widely used language in the world today and Tamil is a Dravidian language spoken predominantly by people of South India and North-East Sri Lanka. Thus considering the importance of these two languages, we chose to develop this MT system.

1.2 PROBLEM DESCRIPTION

Given a grammatically correct English sentence as input, the system should output a Tamil sentence in which the meaning of the original input sentence is preserved. The output Tamil sentence should obey the grammar of the language and convey important information like tense, number and gender as in the original input. The system should also translate documents/paragraphs sentence by sentence.

1.3 SCOPE

There are over 200 countries in the world each having their own official and native languages. The need to communicate at a global level emphasizes the importance of translation. Be it a literature, an article, a technical document or books of some kind, we often find the need to translate between languages. Machine translation also has its application in Cross Lingual Information Retrieval (CLIR). The number of translators available from English Tamil is very less and hence this sys-

tem would be a good contribution to the field of MT.

1.4 CONTRIBUTION

This system is the first English Tamil translator developed using UNL. The standard representation of UNL has been modified to help aid Deconversion to Tamil. Many new attributes have been added for this purpose. This system also best demonstrates the applications and efficiency of UNL. This system is a language generic approach to Machine Translation which is the greatest advantage of using UNL. It helps translate small stories, documents etc. from English to Tamil and can be used by anybody across the world. The readability of the output sentence shows the efficiency of the Morphological generator we have developed. We have handled all the 4 forms (simple, perfect, continuous, perfect continuous) of the 3 tenses (present, past and future). This system shows the efficiency and simplicity of using UNL for translation, thus making a contribution to the applications of UNL. It is evident from the results that this system can easily be developed into one of the most efficient systems for English Tamil translation.

1.5 ORGANISATION OF THESIS

Chapter 2 discusses the existing approaches to MT in greater detail. It also analyses the advantages and disadvantages of each approach. Chapter 3 gives the requirements analysis of the system. It explains the functional and non-functional requirements, constraints and assumptions made in the implementation of the system and the various UML diagrams. Chapter 4 explains the overall system architecture and the design of various modules along with their complexity. Chapter 5 gives the implementation details of each module, describing the algorithms used. Chapter 6 elaborates on the results of the implemented system and gives

an idea of its efficiency. It also contains information about the dataset used for testing and other the observations made during testing. Chapter 7 concludes the thesis and gives an overview of its criticisms. It also states the various extensions that can be made to the system to make it function more effectively.

CHAPTER 2

RELATED WORK

This chapter gives a survey of the possible approaches to Machine Translation. We proposed to work with English to Tamil translation system and we wanted to adopt a language generic approach which would make it easier to extend this system in the future. The output sentence should be grammatically correct and convey the entire meaning of the original input sentence. Thus this survey helped us analyse the various existing approaches to MT and decide on the one which would best cater to our needs. As discussed in Chapter 1, figure 2.1 gives a broad classification of the approaches to machine translation.

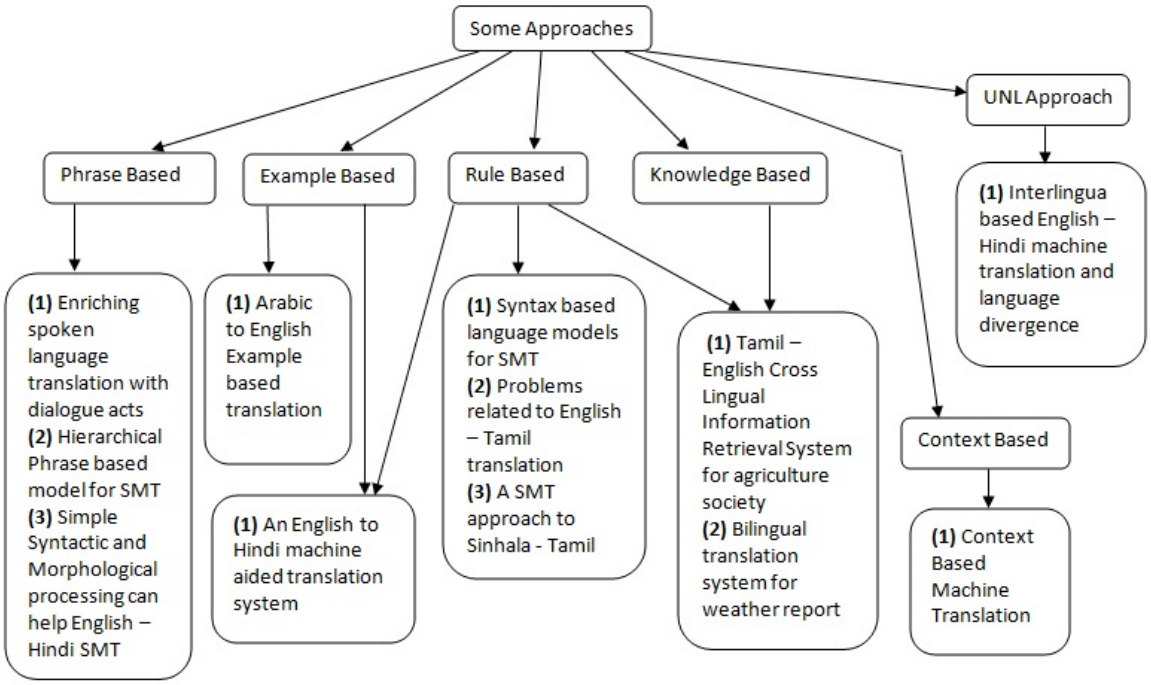

Figure 2.1 A survey of the approaches to Machine Translation

2.1 PHRASE BASED MACHINE TRANSLATION(PBT)

At the outset, this approach involves identifying phrases in the source language text and translating them to target language, which certainly gives better results than a word-to-word alignment. Segment, Reorder and Translate are the three main steps involved in a general phrase based translation[8]. Here phrases can even be a group of words (substring), not necessarily syntactic phrases. A simple PBT system has been discussed by Richard Zens, et al.[32]. Many additional features have been added to this simple system to improve the translation quality[25][21].

Phrase Based Translation is improved by adding dialog acts as they help to identify the nature of the sentence. This is mainly used to depict 'how' the information is conveyed, in addition to 'what' is being conveyed. A Dialogue Act (DA) Tagger has been used to classify the type of sentence. Given a sequence of utterances, each utterance is assigned one of the DA labels. Phrase Based translation is performed first which is then enriched using Dialog Acts[25]. The target translated text is determined using

$$\begin{aligned} T_t^* &= \underset{T_t}{\operatorname{argmax}}\, P(T_t|T_s, L_s) \\ &= \underset{T_t}{\operatorname{argmax}}\, P(T_s|T_t, L_s)\,.\,P(T_t|L_s) \end{aligned} \tag{2.1}$$

In equation 2.1,

- L_s: Source language dialog acts
- T_s: Source alphabet sequence
- T_t: Target alphabet sequence

The first term is a Machine Translation model which is specific to the Dialog Act and the second one is the Language Model. This approach has been experimented for Farsi English, Japanese English and Chinese

English. The major disadvantage here is that DA tagging might involve errors which do not provide the expected enrichment. An improvement of +1 to +4 was observed in the BLEU scores of PBT with DA tagging when compared with the BLEU scores of PBT without DA tagging.

Another improvement to the standard PBT was made by introducing hierarchical phrased based model. Since longer phrases decreases performance, these phrases are logically reordered[8]. Thus in this approach, the reordering of words in a standard phrase based model is extended to reordering to phrases. Rules have been written for this reordering using Context Free Grammar (CFG) and can be learnt dynamically. This has been implemented for Mandarin to English and an improvement of 7.5% on BLEU scores over normal Phrase Based Translation was observed by the proposers of this approach[8].

It has been observed that incorporating syntactic and morphological features the performance of a simple phrase based translation model can be improved for machine translation[21]. The PBT model used here incorporates learning to reorder words in a phrase according to target language syntax. It also uses the suffix of target language words.

- Syntactic information: changing SVO order to SOV order
- Morphological information: a morphological analyzer for source language and a suffix separation program (stemmer) for target language

Syntactic preprocessing is found to reduce the distortion load on standard PBT and morphological preprocessing is mainly used to reduce the training set and also it improves translation to a great extent in morphologically rich languages like Hindi[21]. This system has been implemented for English Hindi translation and 10% to 46% improvement in translation fluency was observed. Ramanathan, Ananthakrishnan et al.[21] claim that substantial improvement in translation quality can be

observed only if a sophisticated morphological analyser is used. Certain other improvements in PBT have been proposed by Richard Zens and Hermann Ney[33].

A new approach to PBT using a pivot language (intermediate language) has been proposed by Hua Wu and Haifeng Wang [31]. Pivot language helps to use PBT to translate between languages for which bilingual corpora is not available. They have concluded that using more pivot languages leads to a better translation quality.

2.2 EXAMPLE BASED MACHINE TRANSLATION(EBMT)

Example based MT is aimed at emulating the way humans learn and do translation. It is usually done by finding similar pieces of text at the word, stem or lemma level. Hence it requires a large bilingual corpus (parallel corpora) to do the translation. As the World Wide Web (WWW) can be thought of as a huge corpus of data, using it as a resource for EBMT is a sensible idea[11]. Initially the examples were stored at paragraph level and they were subjected to morphological analysis and POS tagging. A lookup table matching words from one language to the other was also created. A typical example based approach to machine translation involves

- Matching: Searching the corpus for matching fragments of source text (at least 2 adjacent words). Match scores were computed and certain fragments having score below a threshold were eliminated. Two more approaches to matching other than the canonical ones have also been discussed[19].
- Transfer: Extract the corresponding fragment in the target language (word by word mapping). Sometimes modifications might be needed to words matched at stem or lemma level. Remove

unnecessary words and phrases and compute Total Score as

$$TotalScore = TranslationScore \times MatchScore \tag{2.2}$$

- Recombination: Put together all translated fragments and find N best recombinations that cover the entire input sentence

This approach was used for Arabic English and a BLEU score of 0.1849 was obtained[2]. Kfir Bar, Y. Choueka and N. Dershowitz[2] conclude that the biggest drawback with this system is the necessity to handle many exceptions separately and complicated situations might require new rules. EBMT has also been discussed by Harold Somers[24].

Applying EBMT to short phrases using the context equivalence principle that avoids word to word alignment has also been proposed[28]. Difference between hybrid and pure EBMT has been described by Daniel Jones[15] and has adopted non-hybrid EBMT over hybrid EBMT. Yves Lepage and Etienne Denoual[16] have also explained a pure EBMT. They claim that a pure EBMT works only by using proportional analogy. However they have also studied the effect of dictionaries and paraphrases on the performance of the system. More recent advances in EBMT can be found in the work of Michael Carl and Andy Way[5].

2.3 RULE BASED MACHINE TRANSLATION(RBMT)

This approach to machine translation involves using the rules of the source and target languages to generate target language sentences. A translation model was built by using syntax based language models in SMT. The process begins with constructing, a parse tree of the target language. Then three types of operations were performed on each node of the tree

- Reorder: Changing the order of child nodes according to the rules of the source language used

- Insert: Insert an additional word at each node
- Translate: Translate leaf nodes from target to source language

This has also been extended to cover phrasal translations. As a next step, a decoder (similar to a bottom up parser) was used which would build a decoded-tree from the source language sentence. The decoded tree is such that from this tree we can obtain the target language text just by reordering the leaf nodes and removing source language words in the leaf nodes.

Thus rules of both the languages have been incorporated to build the trees. Therefore given a source language sentence, it builds the decoded-tree from which it can map to the target parse tree and thus obtain the target language sentence. Chinese English translation has been done using this method[6]. Eugene Charniak, Kevin Knight and Kenji Yamada state that this method fails for certain cases which can be handled by incorporating knowledge sources. Another approach to RBMT using chart parsing has been proposed by Andreas Zollmann and Ashish Venugopal[34]. They state that their work of parsing, directly operate at the phrase level in constrast to representing syntactical information in the decoding process.

A basic SMT approach was tried for Sinhala Tamil translation[30]. This approach uses a language model and a translation model. The language model was built using a bilingual corpus and CMU toolkit. The translation model was also built using GIZA++ toolkit using the parallel corpora. The translations returned by the system were scored and best one was selected. BLEU score of 0.185 was produced by this system. But the efficiency of translation largely depends on the corpus being used and its size.

M.B.A.Salai Aaviyamma and Dr.K.Kathiravan[1] have described the process of translation as follows:

- Tokenisation: source language sentence separated into words
- Parsing: root words are identified
- Word Mapping: root words mapped to corresponding words in target language
- Sentence structure changing: rules of target language incorporated here

Problems created when adding tense markers, case markers and while translating proper nouns were handled here. However M.B.A.Salai Aaviyamma Aaviyamma and Dr. K. Kathiravan[1] conclude that complete translation cannot be done only with the rules of the language and requires training of the system. Some techniques which help in word sense disambiguation in RBMT by using contextual information (relationship between a word and its neighboring words) has been suggested[7].

2.4 CONTEXT BASED MACHINE TRANSLATION(CBMT)

The basic principle behind CBMT is to create enormous number of long N-grams which contain a lot of scope for word and phrase based translations. A bilingual dictionary, source language corpus and target language corpus are maintained. As a first step n-grams are identified from the given text, sometimes limited by the number of words. Then, the words/phrases in the n-grams are translated to target language with the help of bilingual dictionary and target corpus. Word order may differ in the target language. Multiple n-grams may be generated in the target language for a single n-gram of the source language. These n-grams are then put together using a decoder to create the target language text.

The translation process begins by splitting the source text into long n-grams which might be overlapping. Then from these overlapping segments, the final target lattice is obtained by maximizing these overlaps

to generate long N-grams. Then the translation of these n-grams to target language involves searching the corpus, finding a list of all possible alternatives and ranking them. Finally a decoder is used to produce the translated output. An approach to evaluate CBMT has been discussed[13].

CBMT is close to example based approach earlier and does not require a parallel corpus. It is suitable only for those languages which are more context dependent than being rule dependant[3]. Spanish to English, Arabic to English and Chinese to English translations have been experimented using this technique[3]. BLEU score analysis shows that, by having a complete dictionary and a larger target language corpus, a score of 0.6950 can be reached[3]. Thuy Linh Nguyen and Stephan Vogel[18] give a more detailed insight into the CBMT technique for Morphological Analysis of Arabic. Any translation which involves using corpus is generally restricted to the words/phrases available in the corpus. For a completely new sentence, not available in the corpus, translation will be quite poor and hence requires modification to the algorithm.

2.5 HYBRID MODELS

All the approaches that have been discussed above have advantages and disadvantages irrespective of the choice of language. In order to overcome this, a combination of approaches has been suggested by M. Tynovsky[29]. Some of the possible combinations proposed by M. Tynovsky[29] are

- SMT affected by EBMT
- SMT affected by RBMT
- EBMT affected by SMT
- EBMT affected by RBMT

- RBMT affected by SMT
- RBMT affected by EBMT

A few techniques to MT falling under the first two categories have also been discussed[29].

2.5.1 Combining Rule based and Example based Approaches

R. M. K. Sinha and A. Jain[23] have proposed an hybrid approach in which example based approach was first applied to translate frequently used words and phrases and then if the former was not successful i.e. exact words or phrases not found in the corpora being used, rule based translation was done. In case of rule based system, a corpus of rules in the form of CFG, a multi-lingual language dictionary and target text generators were used. Word Sense disambiguation was also used to choose the right translation in case multiple options were available. The rule based system first converts the source language text into a 'pseudo target' (an intermediate representation in which most of the ambiguities is resolved) from which translated text is generated using target text generators. This was experimented for English Hindi and was found to generate 90% acceptable translation in sentences which contain atmost 20 words. R. M. K. Sinha and A. Jain state that this approach requires an additional corrector for ill formed sentences and also a human engineered post editing package. An approach to translation involving this hybrid system has also been explained[4].

2.5.2 Combining Rule based and Knowledge based Approaches

This model helps to improve rule based translation by incorporating learning[27]. This kind of hybrid approach is used in Cross Lingual Information Systems. The process involves the following steps:

- Morphological Analyser: Perform a dictionary lookup to see if

the query is directly present. If not, root word is obtained by morphological analysis

- Dictionary Lookup: The result of each intermediate step in morphological analysis is compared with the words in the dictionary until a suitable match is found.
- Machine Translation: A rule based translation is performed (SOV to SVO order). POS tagging is also done here.
- Word Sense Disambiguation: Wordnet is used to find all possible senses of the given word. Each sense is again compared with the senses of all other words in the query and they are scored. The sense that has the highest score is considered the target word.

Dynamic learning module is also incorporated to learn new words. This approach has been implemented for Tamil English CLIR system for Agricultural domain[27]. It produces better results mainly because it is domain specific and uses only specific words.

This approach has also been used in Bilingual Translation System for Weather Report[22]. Rule based approach is directly used for simple sentences and for complex ones Knowledge based approach followed by Rule based approach is used. In case of rule based approach, the sentence pattern is analysed and for each sentence pattern the corresponding pattern of translated text is stored in the rule base. In knowledge based approach, the complex sentence is split into many simple sentences and rule based approach is applied to each of them.

- KBMT process:
  Source language text $\rightarrow$ Tokenization $\rightarrow$ Tagging $\rightarrow$ Lemmatization/Chunking $\rightarrow$ Source language sentence generator $\rightarrow$ Simple sentences
- RBMT process:
  Source language text $\rightarrow$ Morphological Analyser $\rightarrow$ Sentence

type analyser → Mapping to target language

The results of this approach are satisfactory for a small domain. More rules are to be added to make the system more efficient. The precision of Tamil to English translation is less using this system due to the free word order of Tamil language[22].

Several exclusive KBMT approaches have also been used for machine translation. Ghulam Rasool Tahir, et al.[26] have proposed English to Urdu translation using KBMT. The KBMT-89 project at Carnegie Mellon Universitys Center for Machine Translation is discussed by Sergei Nirenburg[20]. This system is used to translate between English and Japanese languages. From the results of these systems we can conclude that hybrid approaches reveal better results than exclusive KBMT approach.

2.6 UNL APPROACH

Universal Networking Language is a declarative formal language specifically designed to represent semantic data extracted from natural language texts[12]. Unlike natural languages, UNL expressions are unambiguous and are composed of Universal Words, Attributes and Relations.

- Universal Words represent simple or compound concepts. They may also contain constraints which restrict the domain in which they are used.
- Attributes provide information about how the concept is being used in the particular sentence. This includes tense, speaker's attitude (affirmative, exclamation etc.), speaker's focus etc.
- Relations are binary relations which explain the relationship between two Universal Words. Eg: obj (object), plc (place), agt (agent) etc.

The process of conversion from any language to UNL is called Enconversion and the reverse process is called Deconversion. This process of Enconversion and Deconversion combined together can be used for translation by using the Enconvertor of source language and Deconvertor of target language[9] with some additional process. UNL structure can be used to derive a lot of inferences for machine learning applications[17].

Using this approach, an English Hindi translation system has been designed[9]. Infact, this is a hybrid of Rule and Knowledge based approaches. This system uses three components namely Hindi analyser to convert Hindi sentences to UNL expressions, English analyser to convert English sentences to UNL expressions and Hindi generator which generates Hindi sentences from UNL expressions. Analysers use Enconvertor and generators use Deconvertor. The differences between the UNL expressions produces from Hindi and the corresponding English sentences were studied and necessary mapping was introduced. For instance, differences were found in Number, Person, Gender, Tense and many other divergences. Separate blocks were introduced to handle these divergences[9].

Some of them were handled by the analysers and for lexical-semantic divergence L-UW dictionary was introduced[9]. L-UW dictionary is the one which gives the corresponding word in the target language. Efficiency of 95% was obtained using this approach. However word sense disambiguation modules need to be introduced in the analysers.

2.7 OBSERVATIONS FROM THE SURVEY

We propose to work on English to Tamil translation system. Since Tamil is a free word order language, no single approach mentioned

above will solve the purpose and so we need to go in for a hybrid approach. Moreover English sentences follow SVO pattern and Tamil sentences follow SOV pattern[27]. Hence it is necessary to incorporate the rules of the language in translation. It was also inferred that incorporating syntactic and morphological information produces better results in translation. Corpus based methods like Example and Context based approaches do not give satisfactory results since the efficiency and fluency of the translation is largely dependent on the corpus being used. The quality of translation decreases greatly if the desired word/phrase is not available in the corpus. Thus to incorporate learning, Knowledge base is essential. Hence we decided to go ahead with a hybrid approach of Rule and Knowledge based systems.

We also wanted to move towards designing a language generic approach for translation by converting the source language to an unambiguous intermediate representation from which we could convert to any target language desired. Thus we chose the UNL approach which involves Enconversion and Deconversion processes. Therefore we aim to implement an Enconvertor for English to UNL and a Deconvertor for UNL to Tamil, also incorporating the other modules that will be needed. Hindi is a partial free word language where the free word order is between the adjacent words of a sentence or a phrase. Tamil is also considered as a free word order language in which sentences normally follow subject, object, verb (SOV) pattern but not necessarily[10]. This makes it clear that the rules would differ for the Deconverters of both languages and so the system developed for English to Hindi translation requires changes in certain modules to adopt it for English to Tamil.

UNL is the best approach towards designing language generic systems. Having Enconvertors and Deconvertors for multiple languages, we could perform translation between 2 languages just by bridging the

gap between their UNLs. Some of the issues in translation such as word sense disambiguation, preserving the meaning of the sentence and grammatical correctness of the translated sentence are handled by the UNL approach. Thus there is a lot of scope in using UNL for translation.

CHAPTER 3

REQUIREMENTS ANALYSIS

3.1 FUNCTIONAL REQUIREMENTS

The system outputs a Tamil sentence for a given English input sentence. The output sentence should adhere to the following requirements:

- The output sentence should convey the meaning of the original input sentence
- The output sentence should be grammatically correct
- The Tamil words in the output sentence should not have any spelling mistakes
- The system must be optimized for time and space complexities
- The system must be able to translate a document/paragraph consisting of many sentences
- The system must be able to translate sentences from any domain

3.2 NON FUNCTIONAL REQUIREMENTS

3.2.1 User Interface

There must be a simple and easy to use user interface where the user should be able to enter his input sentence(s). The intermediate UNL and also the output tamil sentence should be displayed in the screen.

3.2.2 Hardware

No special hardware interface is required for the successful implementation of the system.

3.2.3 Software

- Operating System: Linux
- Programming Language: JAVA
- Database: MySQL
- Tools: NetBeans IDE, Stanford POS tagger, JAWS API

3.2.4 Performance

The system must be optimized, reliable, consistent and available all the time.

3.3 CONSTRAINTS AND ASSUMPTIONS

3.3.1 Constraints

- The system would work only for those words in the dictionary. There are around 1,00,000 English words and 500 tamil words. More tamil words can be added for better results.
- The UNL generated would be correct only for those sentence types for which rules have been written. For other types of sentences, the translation would be incorrect. However, the system can be easily made to work for those by just adding a few rules.
- The accuracy of the POS tagger determines the efficiency of the translation. The errors in the POS tagger would propagate down to the rest of the modules also.

3.3.2 Assumptions

- The input sentence is assumed to be grammatically correct.
- The input is assumed to be either an assertive sentence or an imperative sentence.
- The input sentence does not have any spelling mistakes.

3.4 SYSTEM MODELS

3.4.1 Use Case Diagram

3.4.1.1 Overall Use Case Diagram

The overall usecase diagram of the entire system is shown in figure 3.1. It consists of Enconversion and Deconversion processes.

Pre condition: An English sentence is given as input by the user

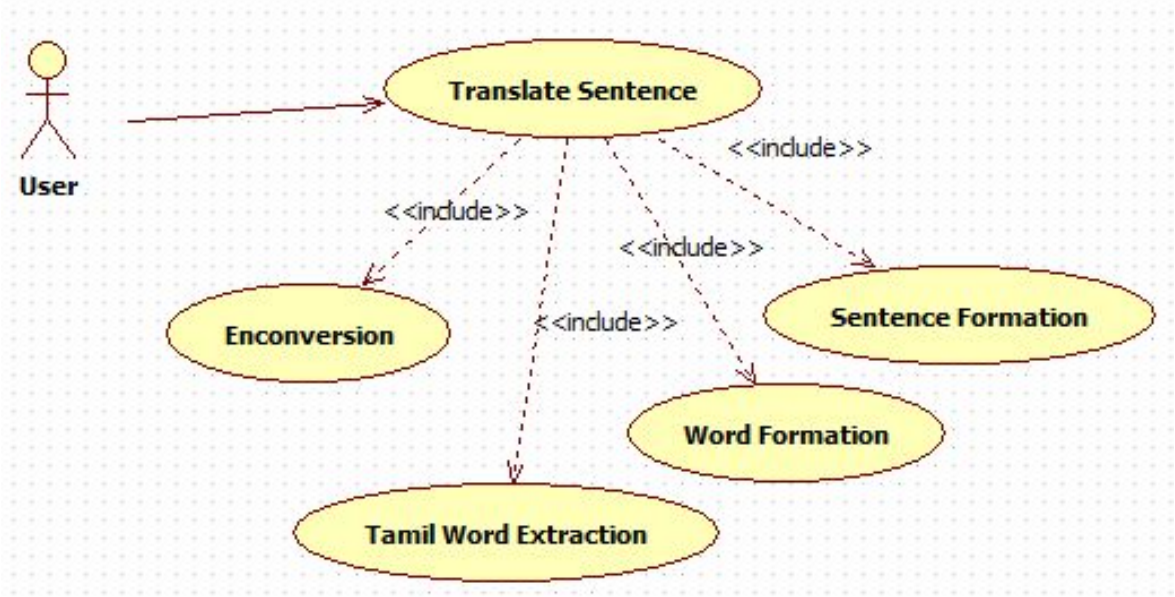

Figure 3.1 Overall Use Case Diagram

Post condition: An equivalent Tamil sentence preserving the meaning and which is grammatically correct is generated.

3.4.1.2 English to UNL Enconversion – Use Case

The usecase diagram of the Enconversion process is shown in figure 3.2. It consists of identifying POS, root words and forming UWs.

Description:

The user gives an English sentence as input. This sentence is converted to an UNL expression using an Enconverter. The UNL Expres-

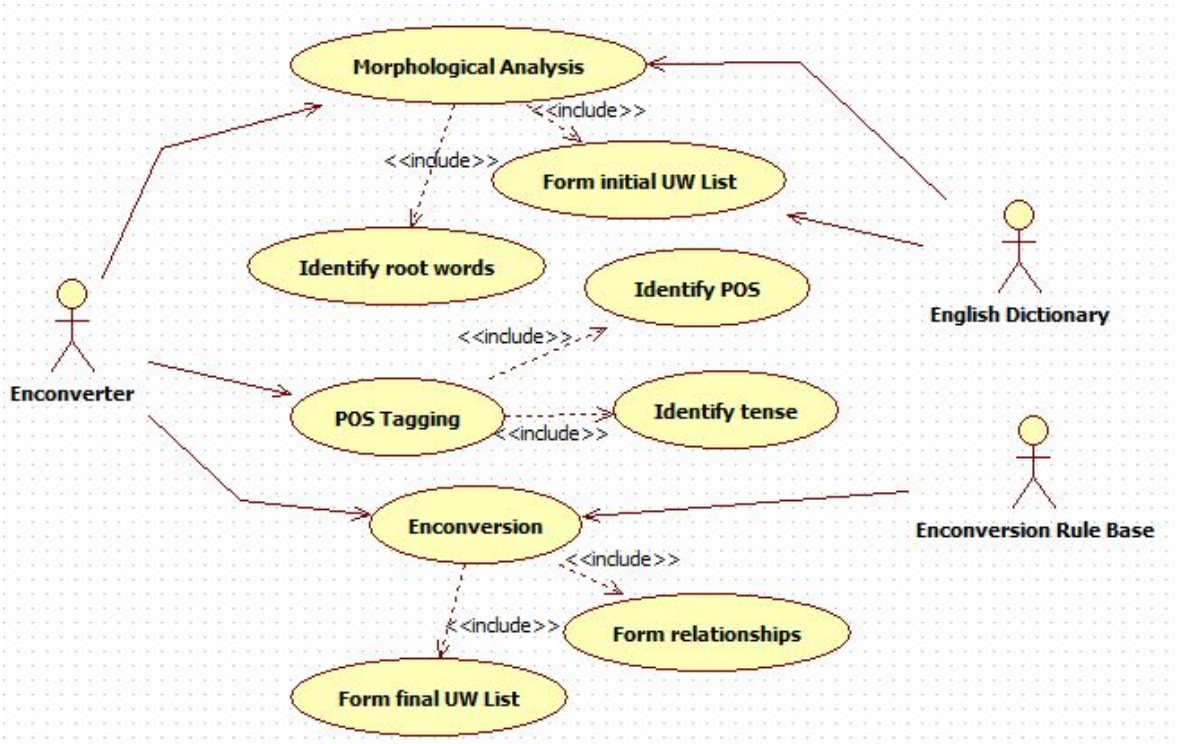

Figure 3.2 Enconversion Use Case Diagram

sion identifies the various concepts, attributes and relationships in the sentence. It also provides gender (male, female), person (first, second, third), number (singular, plural), tense (present, past, future) and various other information. Enconverter uses the English language dictionary and grammar rules.

Pre condition: An English sentence is given as input by the user

Post condition: An UNL Expression is obtained which contains information about gender, person, number, tense, various concepts, relationships between concepts etc.

3.4.1.3 UNL to Tamil Deconversion – Use Case

The usecase diagram of the Deconversion process is shown in figure 3.3. It consists of extracting tamil words, forming complete verbs and nouns and formation of the final Tamil sentence.

Description:

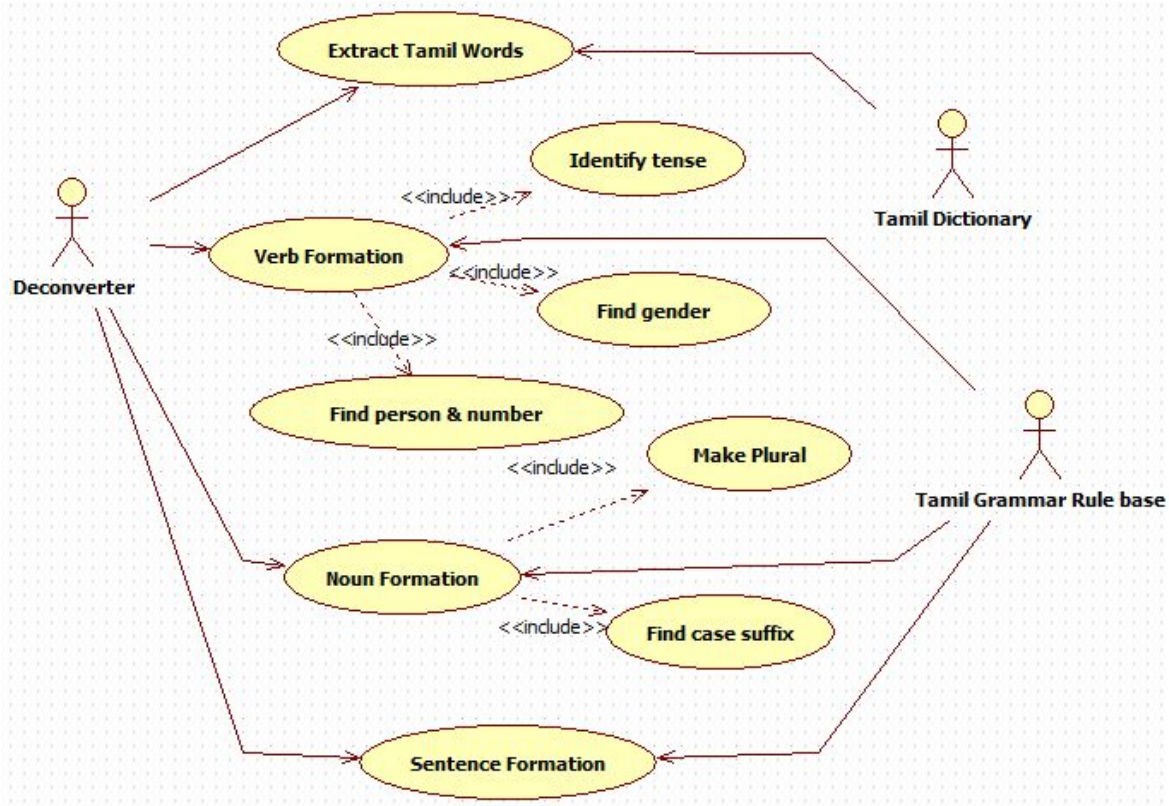

Figure 3.3 Deconversion Use Case Diagram

Once the UNL Expression is generated, the required Tamil sentence is obtained from it by processing the UNL Expression. The UNL Expression is analysed for relationship between different UWs. Person, Number and Gender information are extracted. Tamil words (only root) for the English UWs are extracted from the dictionary. Complete Tamil words (especially Verb and Noun) are formed by adding appropriate suffixes using grammar rules. Finally the required Tamil sentence is generated with the help of rule base.

Pre condition: An UNL Expression is generated from the input English sentence.

Post condition: A meaningful Tamil sentence equivalent to the input English sentence and which is grammatically correct is obtained.

3.4.2 Sequence Diagram

3.4.2.1 Enconversion – Sequence Diagram

The sequence of steps involved in the process of Enconversion is shown in figure 3.4. It can be seen that Morphological Analyser, POS Tagger and Enconverter are the three main components involved. The interaction between these components are shown in detail in figure 3.4.

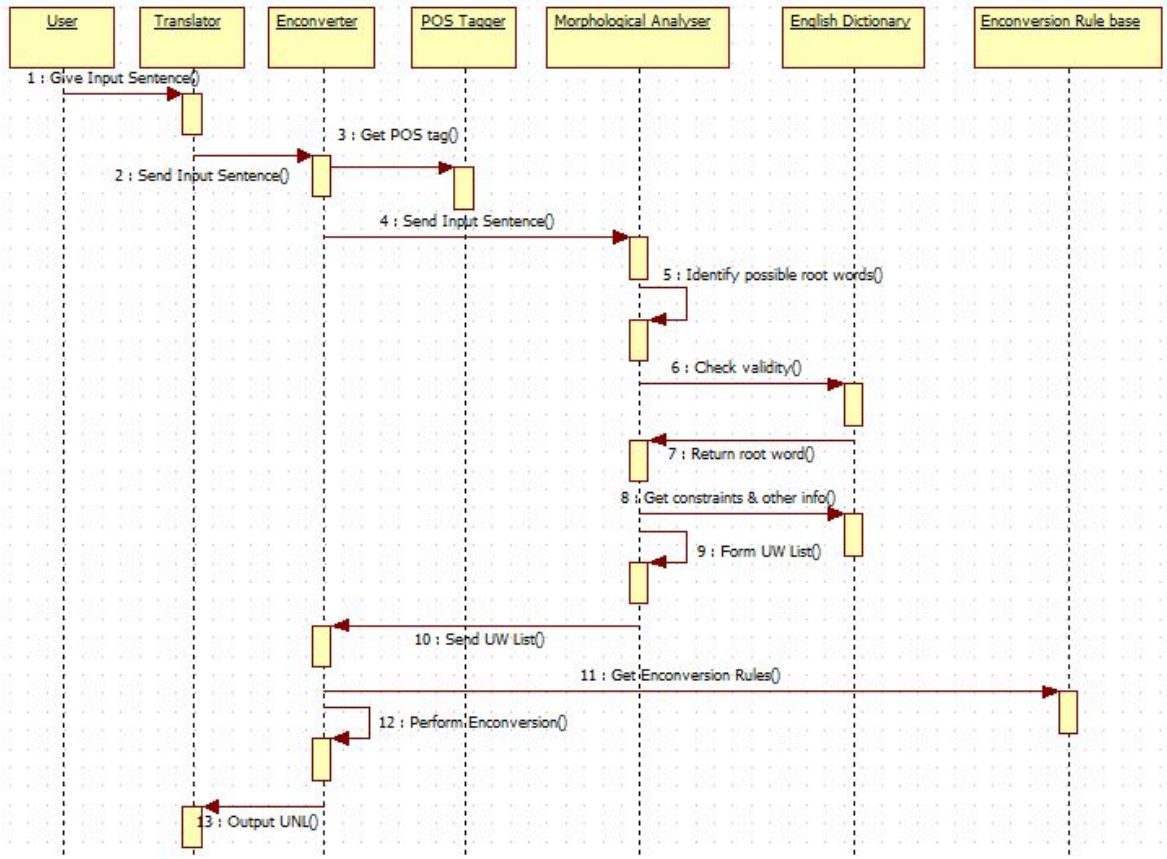

Figure 3.4 Sequence Diagram for Enconversion

3.4.2.1 Deconversion – Sequence Diagram

The sequence of steps involved in the process of Enconversion is shown in figure 3.4. It can be seen that Tamil Word Extractor, Morphological Generator and Sentence Formation are the three main compo-

nents involved. The interaction between these components are shown in detail in figure 3.5.

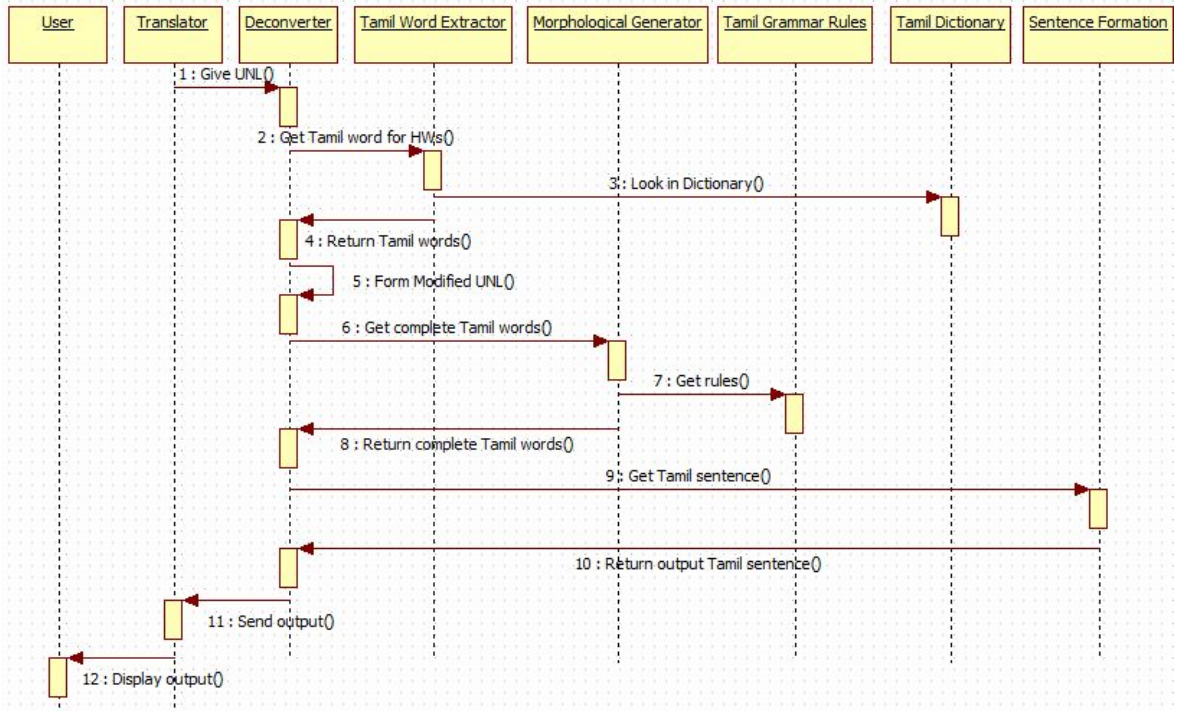

Figure 3.5 Sequence Diagram for Deconversion

CHAPTER 4

SYSTEM DESIGN

4.1 SYSTEM ARCHITECTURE

The block diagram of the entire system is shown in figure 4.1. The Morphological Analyser has been implemented using JAWS API. UNL Knowledge Base, UNL Enconverter and the Envonversion rule base have been developed. A Morphological generator for Tamil has been developed and an algorithm for sentence formation has been devised.

The system aims at translating a given English sentence to a Tamil sentence which conveys the meaning of the input by ensuring a grammatically correct sentence as output. The UNL Wordnet Dictionary developed by Ronaldo Martins was used. Tamil words were manually inserted into the dictionary. The UNL Knowledge Base was created by parsing through the constraint list in the dictionary. The input English sentence is given to the POS tagger and the Morphological Analyser. The POS tagger returns the part-of-speech of every word in the sentence and the Morphological Analyser returns a list of possible root words. These root words are checked with the dictionary to obtain a meaningful English word as the root. The Dictionary is then looked up to obtain the domain constraint and other required attributes. Thus the initial Universal Word(UW) list is formed which contains the Head Word(HW), domain constraint and other attributes.

This list would be processed by the Enconversion algorithm. Rules for Enconversion are written following a standard template, in accor-

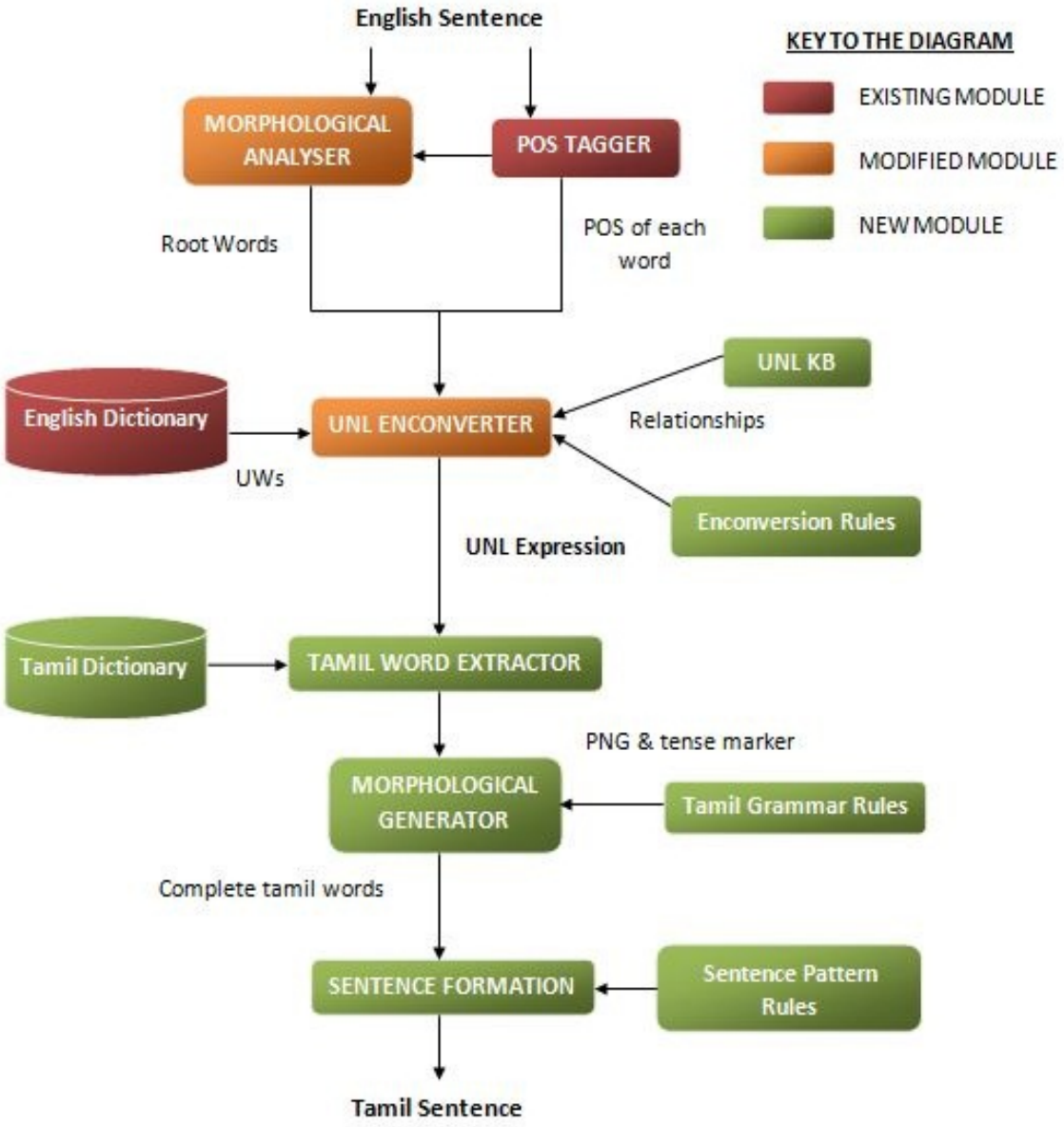

Figure 4.1 System Architecture

dance with the English grammar and the target UNL required. The algorithm uses these rules to form the final UWs of the output sentence and also the relationships between these UWs. This gives the intermediate UNL expression. Then the Tamil word for every Head Word(HW) in the final UW list is obtained from the dictionary by matching its constraint list and POS. These root Tamil words are given to the Morphological Generator which forms the complete Tamil words by adding the required suffixes like tense marker, PNG marker and case suffix marker according to Tamil grammar rules. Finally the newly written rule-based sentence formation algorithm reorders the words to obtain the output Tamil sentence.

4.2 UI DESIGN

A simple and easy to use User Interface has been designed for the system using the Netbeans IDE. The UI contains three text areas for English input, UNL and Tamil Output. The input sentences/paragraph is inserted in the first one. On clicking the Translate button, the intermediate UNL will be displayed in the second text area and the final output will be displayed in the third. The UI also shows the time taken for translation and the BLEU score (an evaluation measure of the output) obtained for the input given by the user. The UI Design of our system is shown in figure 4.2.

4.3 CLASS DIAGRAM

The class diagram of the entire Machine Translation system is shown in figure 4.3. This diagram depicts the functions of various modules in the system clearly. It also shows the interaction between the modules of the system thereby providing a clear idea for implementation.

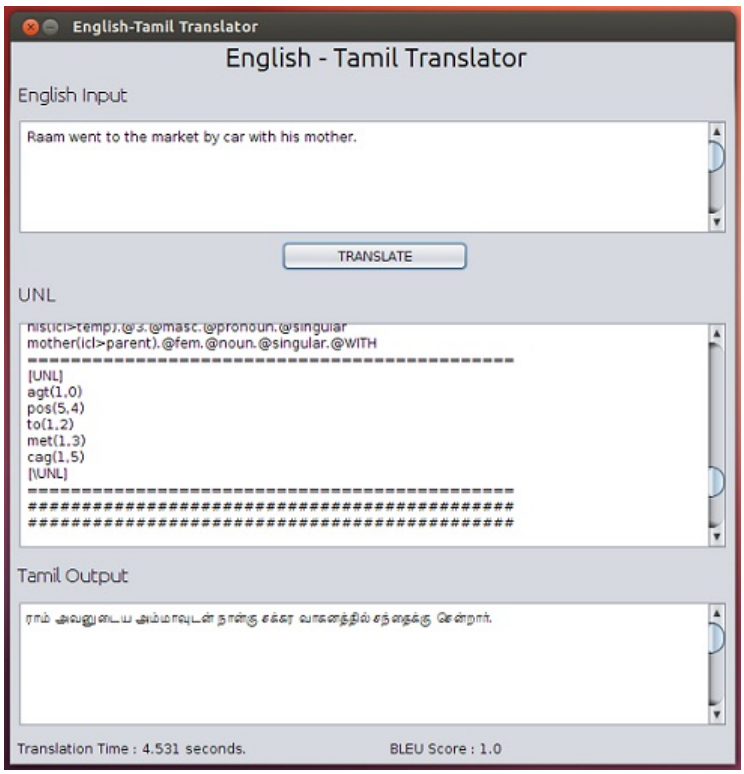

Figure 4.2 UI Design

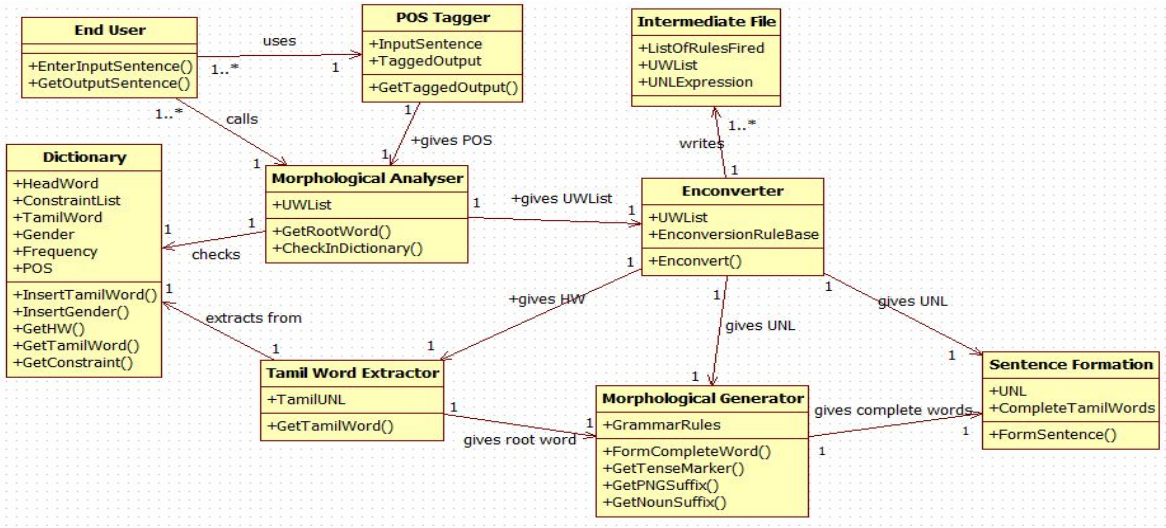

Figure 4.3 Class Diagram

4.4 MODULE DESIGN

4.4.1 English Dictionary

The UNL Wordnet dictionary developed by Dr. Ronaldo Martins, Mackenzie University, Brazil is used. This dictionary consists of about

1,00,000 English words. The dictionary is stored in a MySQL database. The database has been modified to contain Tamil words for UWs and also gender information. Following are some of the important information in the dictionary database.

- Head Word
- Constraint List
- Part Of Speech
- Tamil Word
- Gender (for certain specific words denoting masculine and feminine genders)

Since a single word may occur with different part-of-speech in different contexts, multiple entries would be available for a single word in the dictionary. The dictionary is used by the Morphological Analyser to extract a meaningful root word and also to find the domain constraint of a particular word. The database is huge and querying it consumes a lot of time. Two multicolumn indexes are created in the DB to enhance information retrieval. The multicolumn indexes are created by concatenating the values of the two columns and sorting them in ascending order. Following are the two multicolumn indexes which have been created.

- Head Word and Part Of Speech
- Head Word and Constraint List

4.4.2 POS Tagger

The Stanford POS tagger API has been used in developing this module. The entire input sentence is given to this module which returns the part of speech of every word in the sentence. Stanford POS tagger identifies around 36 different parts-of-speech. This module is very crucial to the development of the system and the errors that occur here penetrate through every other module that follows.

4.4.3 Morphological Analyser

This module has been developed using the JAWS API. Given a word and its POS, a list of possible root words is returned by the API. From this list, the first word that occurs in the dictionary has been selected as the root word. This morphological root forms the Head Word of that Universal Word (UW) in the UNL representation. The Morphological Analyser also forms the initial UW list. A UW consists of three parts namely

- Head Word
- Domain Constraint
- Attributes

Head Word is obtained using the JAWS API and the Constraint List is obtained from the Dictionary by matching the head word and its POS using not null constraint. Attributes are added to the UW as and when required until the process of Enconversion is complete. Following are the various purposes for which attributes are added

- To indicate POS
- To indicate person, number and gender information (both in nouns and in verbs)
- To indicate the presence of a relationship
- To classify the articles and determinants

PNG information is required in verbs in order to generate suffix for the formation of complete Tamil verb. Thus there is a modification from the standard representation of UNL as stated by UNDL.

4.4.4 UNL Knowledge Base

The UNL KB has been created by parsing the dictionary. Thus it contains only inheritance relationships. This can be used to hold information about certain UNL relationships also. The KB is mainly used

to enhance the UNL representation and make the formation of relationships more meaningful thus giving a proper semantic structure.

4.4.5 Enconversion Rule Base

These are rules which aid the process of Enconversion. They are split into two parts as explained below.

1. Rules which merge adjacent UWs to create a final list of necessary UWs that are required in the translated output
2. Rules which form relationships between any two UWs

These two categories of rules are broadly classified as

- Left Composition
- Right Composition
- Left Modification
- Right Modification
- Attribute Changing

Rules have been written following a common template given below.

- **Rule Template:**
  (Specifications of Node-1), (Specifications of Node-2) := (Resultant Node-1), (Resultant Node-2)
- **LHS Template:**
  (HW, ICL, ATTR)
- **RHS Template:**
  $(0|1|2, \quad REL| \quad + \quad ATTR, \quad \%1| \quad - \quad ATTR, \quad \%2| \quad + \quad IND\_ATTR, \; -IND\_ATTR)$

LHS Template: This indicates there can be rules specifying the Head Word itself or its domain constraint or certain attributes of the UW.

RHS Template: Here the first parameter indicates the action taken on the node. '0' denotes deletion of the node, '1' denotes formation of relationship and '2' denotes modification of the existing node. Depending

on the value of this parameter, the remaining parameters convey varied information.

- When first parameter is '0' then all the remaining parameters would be empty.
- When first parameter is '1' (formation of relationship)
  - The second parameter indicates the name of the relationship
  - The third parameter denotes the node which comes first in the relationship
  - The fourth parameter denotes the node which comes second in the relationship
  - The fifth parameter would be empty
  - %1 denotes the first node on LHS and %2 denotes the second node in LHS
- When first parameter is '2' (modification of node)
  - The second parameter denotes the attribute(s) to be added
  - The third one denotes the attribute(s) to be removed
  - The fourth parameter denotes any specific attribute(s) of nodes on the LHS to be added
  - The fifth parameter denotes any specific attribute(s) of nodes on the LHS to be removed

4.4.6 UNL Enconversion Algorithm

The algorithm largely resembles the one followed by UNDL, the difference being we look at only two nodes at a time (ie) there is only one Analysis and one Condition window. There are three rulesets, ruleset-1, ruleset-2 and ruleset-3, which respectively contain the rules that are to be executed in order. Ruleset-1 would be executed first, followed by the other two in order.

Initially the first two nodes are taken. Having two nodes, the

ruleset-1 is checked first to see if any rule matches. If matched, the rule is executed and both the windows move by one position. If no rule is matched, the windows just move. A single iteration is said to have been completed when the entire sentence is parsed once. Ruleset-1 will be checked repeatedly until no rule is matched for an entire iteration thus making several passes through the input sentence.

The algorithm then proceeds to check ruleset-2 in the same fashion, forming relationships. Ruleset-2 will be checked repeatedly until no rule is matched for an entire iteration. In case of simple sentences, the processing ends here and a single node would remain. If more than one node remains, we move on to check ruleset-3 which forms the relationship for compound/complex sentences as a composition of simple sentences. The point at which only one node remains in the UW list, indicates that the UNL graph has been constructed. This node is the first node (start node) of the graph. The order of the rules plays an important role in the proper formation of final list of UWs and also in the formation of relationships.

4.4.7 Tamil Word Extractor

This module retrieves the Tamil word corresponding to every Head Word (HW) in the UNL. Tamil words are inserted into the English dictionary manually. Thus the dictionary is used as UW dictionary. The query to retrieve the Tamil word matches the HW, its domain constraint and POS. Only the root word is extracted here. For HWs whose equivalent Tamil word is not found in the dictionary Phonetic Transliteration has been done as a new component to the algorithm so as to aid translation. English mappings are created for Tamil vowels and consonants. The rest of the Tamil consonantal vowels are mapped by combinations of the above two. Our proposed algorithm chooses the first longest map-

ping during the process of transliteration.

4.4.8 Morphological Generator

This module forms the complete Tamil words for the root words extracted. The two main parts are formation of verbs and formation of nouns. Formation of complete verbs requires the addition of tense marker and PNG suffix to the root verb. Formation of complete nouns requires the addition of plural suffix and case marker. Case markers are added to the nouns depending on the UNL relationship in which they are involved. Apart from these rules, the set of rules which decide the output when two Tamil words combine are also written. For example, these rules are required to determine the output word when suffixes are added to the root.

4.4.9 Sentence Formation

Tamil sentences mostly follow the SOV order[27]. The verbs in the input sentence are considered as central nodes. The subject that lies just before the verb is added to the verb. The phrase (between two verbs) that lies after the verb is reversed and added to the verb. Finally the sentence if formed just by concatenation (ie) the subject comes first, the phrase in reversed order, then the verb and this might be followed by any number of similar patterns. In case of compound/complex sentences, tamil words corresponding to the conjunction like marrum, aanal etc. are inserted.

4.5 COMPLEXITY ANALYSIS

4.5.1 Time Complexity

The time complexity of each module of the MT system is shown in Table 4.1.

Table 4.1 Time Complexity of various modules

S.No	Module	Complexity(per sentence)
1	Morphological Analyser	O(N)
2	POS Tagger	O(NlogN)
3	UNL Enconverter	O(K*N*N)
4	Tamil Word Extractor	O(N)
5	Morphological Generator	O(N)
6	Sentence Formation	O(N)

- 'K' denotes No. of rules
- 'N' denotes No. of words in the sentence

4.5.2 Complexity of the Project

- The complexity of the project lies in expressing the grammar of source and target languages. Rules are to be written to handle every case possible. We have written rules to handle most of the simple English sentences and a few cases of compound sentences.
- The accuracy of the POS tagger also affects the efficiency of the output. Hence an efficient POS tagger needs to be chosen.
- Certain additional information need to be added to the standard UNL representation in order to make it easier to convert it to Tamil. Analysis should be made on the changes required in the UNL representation.
- The efficiency of the Morphological Generator also plays an important role. Morphological generator has been developed from the scratch. Rules for choosing the tense marker, PNG marker and case marker for nouns have been written.
- In addition, rules which decide the output, when two Tamil words combine are also written. Errors in these rules would produce spelling mistakes and other grammatical errors in the output. So

care should be taken in writing these rules.

- Sentence formation is another complex task since there can be more than one possible correct output. We have followed the SOV order keeping verb as the central node.
- The dictionary database is huge and hence accessing it would take a lot of time. So proper indexing needs to be done on the DB.

CHAPTER 5

SYSTEM DEVELOPMENT

The system described consists of various packages like POS Tagger, Morphological Analyser, Enconverter, Tamil word Extractor and Morphological Generator. The overall code overview showing the organisation of these various packages of the Machine Translation system can be seen in figure 5.1.

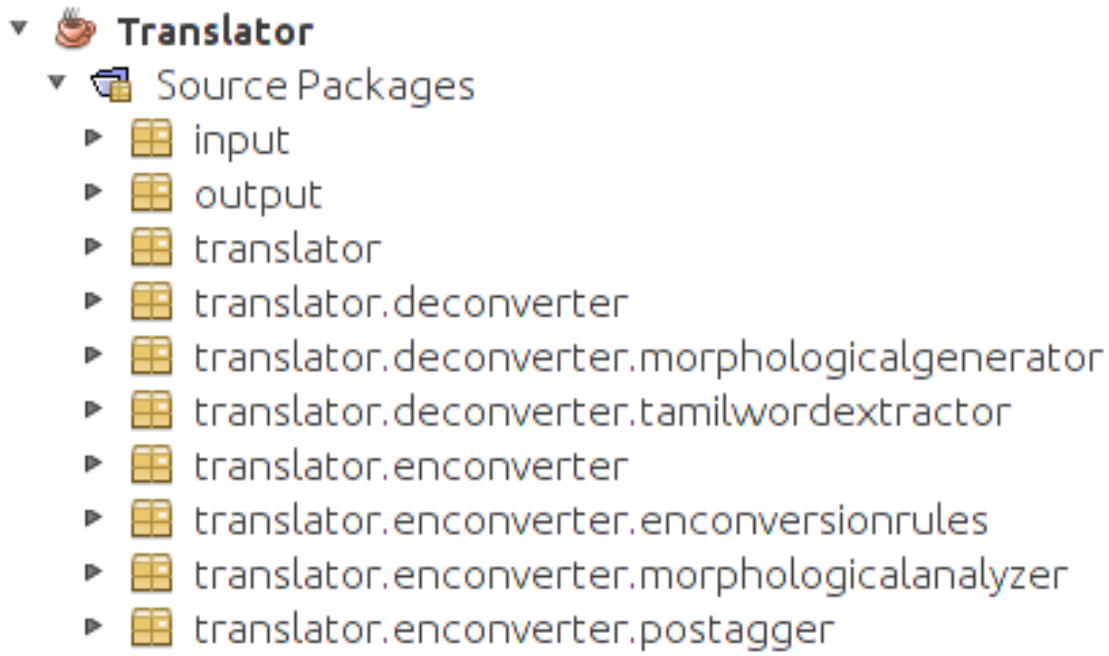

Figure 5.1 Code Overview

An overview of the algorithm of entire system is shown below. The input sentence S is given to the Enconverter which produces the UNL Expression. This UNL Expression is given to the Deconverter which outputs the translated Tamil sentence T.

S ← Input Sentence

TRANSLATE(S)

1. UNL ← ENCONVERT(S)
2. T ← DECONVERT(UNL)

5.1 PROTOTYPE ACROSS THE MODULES

The input and output to each module of the system is described in this section.

- **POS tagger:** This module takes the English sentence as input and gives the part-of-speech of every word in the context of that specific sentence.
- **Morphological Analyser:** This module takes the English sentence and its POS as input and outputs the root word which should be present in the Dictionary. It also adds the constraint list and attributes to every Head Word to form the initial UW List.
- **UNL Enconverter:** The Enconverter operates on the initial UW List created by the Morphological Analyser. Using the Enconversion Rules written, the UW list is processed to form the required UNL Expression. Some of the Enconversion rules are given in Appendix B.
- **Tamil Word Extractor:** It takes the UNL expression as input and replaces the Head Word in every UW with its equivalent Tamil word. If an equivalent Tamil word is not found in the dictionary, then Transliteration is done. Thus it outputs a modified UNL Expression.
- **Morphological Generator:** This module takes the HW, attributes and relationships from the modified UNL and forms the complete Tamil words according to the rules of Tamil grammar. Appendix B gives a sample of the rules used to build the Morphological

Generator.

- **Sentence Formation:** This module processes the complete Tamil words formed by the Morphological Generator according to a new rule-based algorithm and outputs a translated Tamil sentence.

5.2 ENCONVERSION ALGORITHM

The Enconversion algorithm is given below. It outputs the UNL Expression of the given input sentence to an intermediate file.

ENCONVERT(S)

1. n ← No. of words in S
2. W ← Words in S
3. **for** i ← 1 **to** n
4. RW[i] ← MorphologicalAnalysis(W[i])
5. POS[i] ← PosTag(W[i])
6. **for** i ← 1 **to** n
7. C[i] ← ConstraintList(FindInDic(RW[i],POS[i]))
8. Attr[i] ← Attributes derived based on POS[i]
9. **for** i ← 1 **to** n
10. UW[i] ← RW[i] + C[i] + Attr[i]
11. UNL ← ”[UNL]”
12. R_1← General Rules
13. k_1← No. of general rules
14. R_2← Relationship Rules
15. k_2← No. of relationship rules
16. R_3← Rules for compound sentences
17. k_3← No. of rules for compound sentences
18. MatchCount ← 0
19. **for** i ← 1 **to** k_1
20. **if**(R_1[i] matches) **then**

21. Execute R_1[i]
22. MatchCount ← MatchCount + 1
23. **if** (Matchcount >0) **then**
24. **goto** 18
25. MatchCount ← 0
26. **for** i ← 1 **to** k_2
27. **if**(R_2[i] matches) **then**
28. UNL ← UNL + Execute R_2[i]
29. MatchCount ← MatchCount + 1
30. **if** (Matchcount >0) **then**
31. **goto** 25
32. MatchCount ← 0
33. **for** i ← 1 **to** k_3
34. **if**(R_3[i] matches) **then**
35. UNL ← UNL + Execute R_3[i]
36. MatchCount ← MatchCount + 1
37. **if** (Matchcount >0) **then**
38. **goto** 32
39. UNL ← UNL + ”[/UNL]”
40. **return** UNL

5.3 DECONVERSION ALGORITHM

The Deconversion algorithm is given below. The output of this algorithm is the translated Tamil sentence T.

DECONVERT(UNL)

1. n ← No. of UWs in UNL
2. **for** i ← 1 **to** n
3. TW[i] ← ExtractTamilWord(UW[i])
4. UW[i] ← TW[i] + Constraints + Attributes

5. **for** i ← 1 **to** n
6. TW[i] ← GenerateTamilWord(UW[i])
7. UW[i] ← TW[i] + Constraints + Attributes
8. T ← FormSentence(UW)
9. **return** T

5.4 DEPLOYMENT DETAILS

The deployment of the system requires Stanford POS Tagger API and JAWS API. Wordnet Database should also be present. The UNL Wordnet dictionary should be available as a MySQL Database. Any IDE like Netbeans can be used to deploy the system successfully.

CHAPTER 6

RESULTS AND DISCUSSION

6.1 DATASET FOR TESTING

The input to the system consists of 264 sentences taken from CBSE English textbooks of classes 1-5. The dictionary has about 350 Tamil words inserted into it. The test data consists of various types of sentences like simple, compound and complex. Each module of the system was also tested separately. The results of this module testing as well as the testing of the entire system are summarized below.

6.2 OUTPUT OBTAINED IN VARIOUS STAGES

This section shows the results obtained during module testing.

6.2.1 Input Sentence

The input sentence shown in figure 6.1 was given to the system.

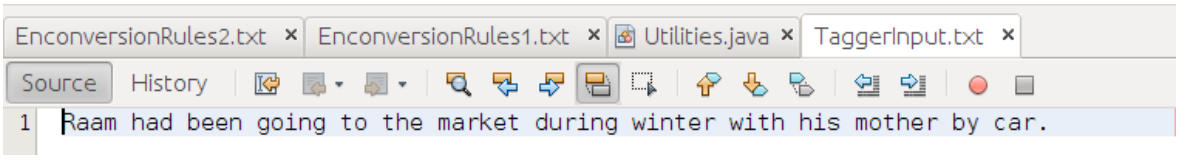

Figure 6.1 Input Sentence

6.2.2 POS Tagger

The output of POS tagger is shown in figure 6.2.

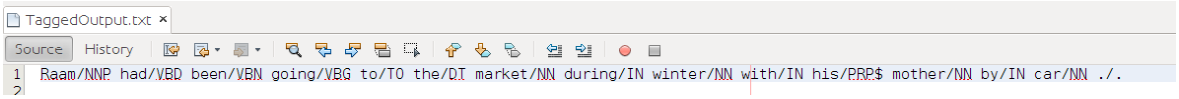

Figure 6.2 Output of POS Tagger

6.2.3 Formation of Universal Words

The Formation of UW is shown in figure 6.3. This is the initial UW list which is the output of Morphological Analyser.

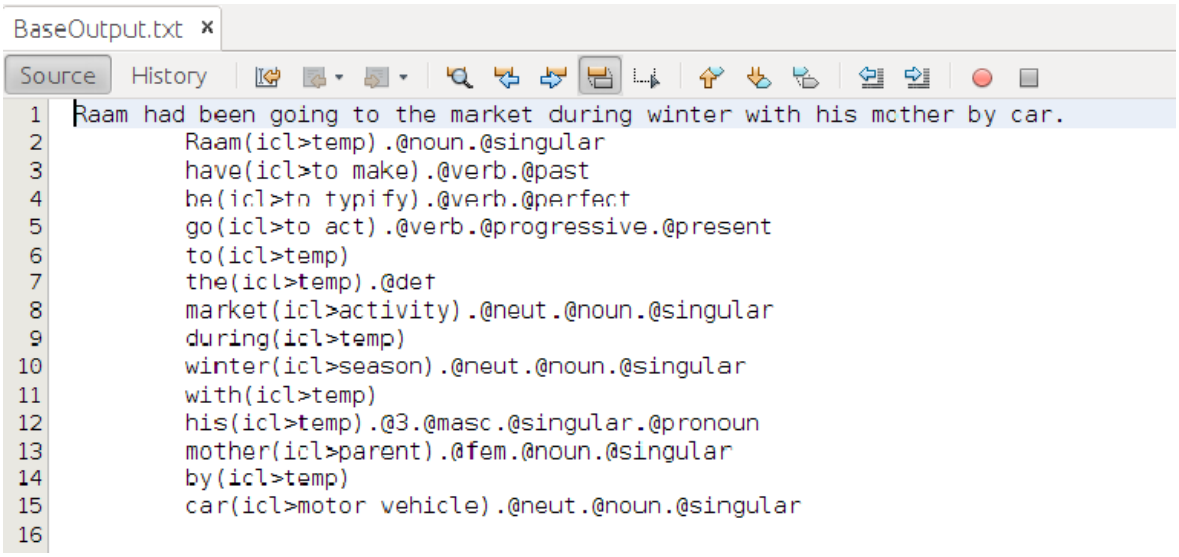

Figure 6.3 Output of Morphological Analyser

6.2.4 UNL Expression

The UNL Expression of the input sentence is shown in figure 6.4. The system also outputs the various stages during the process of Enconversion. The rules that are being fired at each point and how the UWs get combined to form relationships can be explicitly seen.

6.2.5 Tamil Word Extraction

Figure 6.5 shows the UNL that will be formed after the Tamil word is extracted from the dictionary. Every HW in the Universal Word is just replaced with its corresponding Tamil word. The Tamil word is

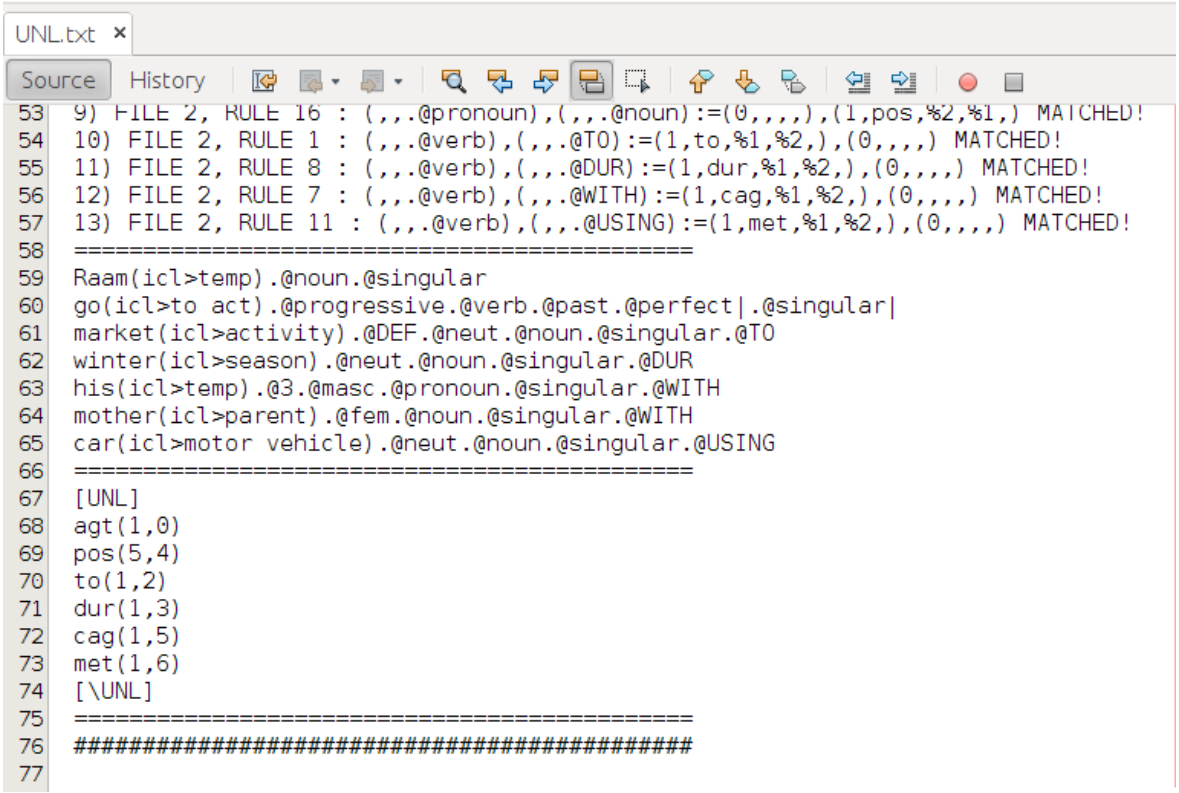

Figure 6.4 Output of Enconverter-UNL

extracted from the dictionary by matching the Head Word, POS and Constraint List of every UW in the UNL expression.

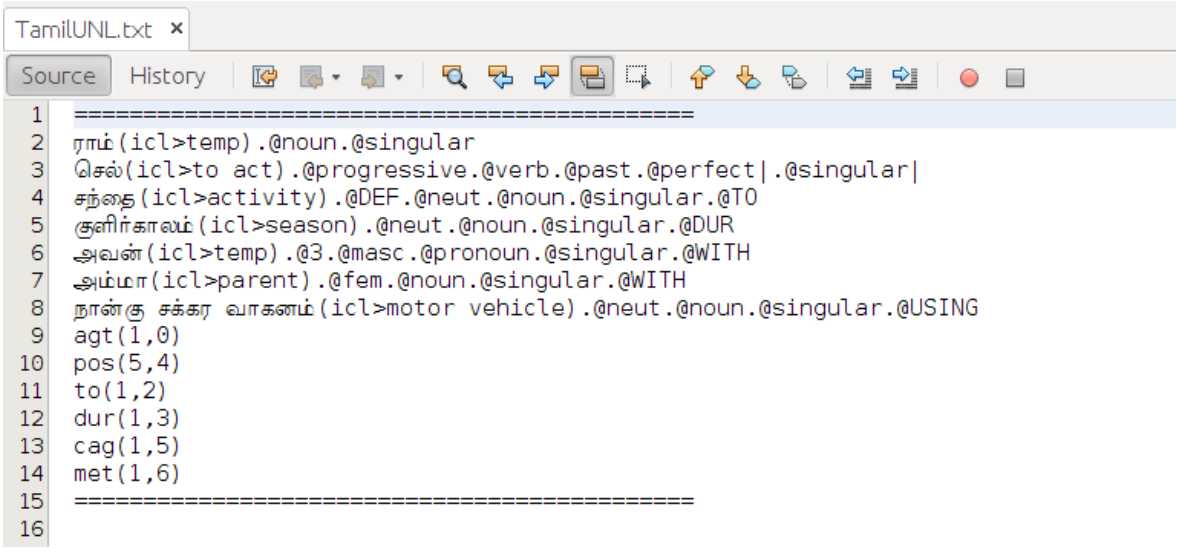

Figure 6.5 Output of Tamil Word Extractor

6.2.6 Morphological Generator

The formation of complete Tamil words using the root words extracted from the dictionary is shown in figure 6.6. This is achieved by deciding and adding the tense marker and PNG marker to Tamil verbs and case suffixes to the Tamil nouns.

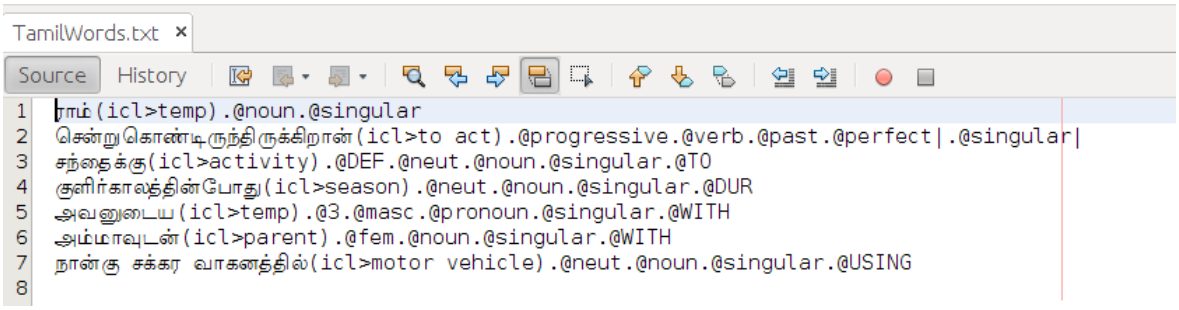

Figure 6.6 Output of Morphological Generator

6.2.7 Output Tamil Sentence

The final output of the MT system after applying the Sentence Formation algorithm is shown in figure 6.7. It is a rule based algorithm which rearranges the words of the input sentence in SOV order. In some cases, the sentence is rearranged completely to enhance readability to convey the exact meaning of the input sentence.

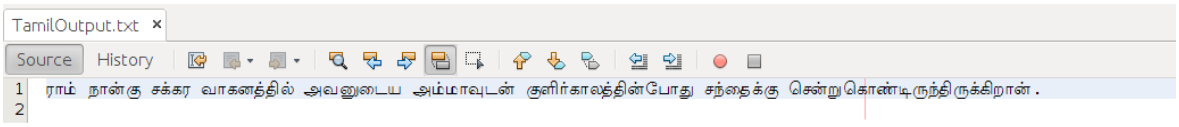

Figure 6.7 Output Tamil Sentence

6.3 SAMPLE SCREENSHOTS DURING TESTING

A part of the input and output sentences during testing are shown in figure 6.8 and 6.9 respectively. The system is tested for various test cases which are detailed in Appendix A module-wise.

My house is red.
I am a happy child.
I laugh and play the whole day.
I hardly cry.
I have a green tree.
I have a red pen.
The sun shines in the sky.
Sonu lived in a house of straw.
Monu lived in a house of sticks.
A bad wolf came.
The wolf destroyed the house.
The house was strong.
I save water.
The wet towel is on the floor.
The hunters went to the forest.
They came to a river.
They did not cross the river.
I can wear shoes.
I can blow bubbles.
A bell fell in the well.
The well was big and the monkey fell.
A fat cat saw a rat.
The fish lives in the pond.
There was a hen.

Figure 6.8 A part of test input

என்னுடைய வீடு சிவப்பாக இருக்கிறது.
நான் ஒரு மகிழ்ச்சியான குழந்தையாக இருக்கிறேன்.
நானுடைய லெளக் முழுதான பகலை விளையாடு.
அரிதாக நான் அழுகிறேன்.
நான் ஒரு பச்சையான மரத்தை வைத்திருகிறேன்.
நான் ஒரு சிவப்பான பேனாவை வைத்திருகிறேன்.
சூரியன் வானத்தில் ஒளிர்க்கிறது.
ஸொனு வைக்கோலின் ஒரு விட்டில் வாழ்ந்தார்.
மொனு குச்சிகளின் ஒரு விட்டில் வாழ்ந்தார்.
ஒரு கெட்டதான ஓநாய் வந்தது.
ஓநாய் விட்டை அழித்தது.
வீடு பலமாக இருந்தது.
நான் தண்ணீரை சேமிக்கிறேன்.
ஈரமான துண்டு தரையில் இருக்கிறது.
வேடன்கள் காட்டுக்கு சென்றார்கள்.
அவர்கள் ஒரு ஆறுக்கு வந்தார்கள்.
அவர்கள் ஆறை கட்டாமல் இருந்தார்கள்.
நான் காலணிகளை அணிவேன்.
நான் நீர்க்குமிழிகளை ஊதுவேன்.
ஒரு மணி கிணறில் விழுந்தது.
கிணறு பெரிதான குரங்காக இருந்தது மற்றும் விழுந்தது.
ஒரு குண்டான பூனை ஒரு எலியை பார்த்தது.
மீன் குளத்தில் வாழ்கிறது.
தெரெ ஒரு கோழியாக இருந்தார்.

Figure 6.9 A part of test output

6.4 PERFORMANCE EVALUATION

The performance of the entire system is evaluated using the standard parameters described below.

6.4.1 BLEU Score

6.4.1.1 Description

BLEU (Bilingual Evaluation Understudy) is an algorithm for evaluating the quality of machine- translated text. It is used to determine how close the machine translated output is to that of a human translated one. Thus the computation of BLEU score requires a good reference translation (human translated text). There can be more than one reference translations to get a more accurate score. The score is calculated for every sentence in the corpus and then averaged to determine the overall translation quality. BLEU is always an output between 0 and 1. Values closer to 1 indicate better translations.

6.4.1.2 Algorithm

The algorithm to compute BLEU is as follows.

- For every word in the candidate translation, count the number of times it occurs in the reference translation.
- Sum the count of all the words in the sentence.
- Divide the sum by the number of words in the reference translation.

The words here correspond to unigrams. Instead of unigrams, longer n-grams can be chosen which determine the fluency of translation better.

6.4.1.3 Adaptation to Tamil

We have used unigrams with one reference translation for the computation of BLEU. Instead of just assigning 0 or 1 according to whether a particular word is present or not in the reference translation, we assign partial scores to every word using Edit Distance. This is done because, there might be minor spelling mistakes in Tamil (like sandhi) in contrast to English, which in reality might not be major mistakes.

- Every word of the candidate translation is compared with every word of the reference translation which has a common prefix of length atleast 3.
- Levenshtein distance (also referred to as Edit Distance) is computed for these two strings (words).
- This Levenshtein distance is summed up and divided by the total no. of words in the sentence.

Levenshtein distance between two words is the minimum number of single-character edits (insertion, deletion, substitution) required to change one word into the other.

6.4.1.4 Score obtained and discussion

We obtained a BLEU score of **0.581** using this methodology. The BLEU score was calculated for the output of the Google translate with the same corpus. The score obtained was **0.3489**. This low score indicates the inefficiency of Example Based approach to MT. Thus UNL approach to Machine Translation is generic and works better than existing approaches. This higher BLEU score is attributed to the efficiency of Enconversion rules and the efficiency of Morphological Generator.

The sentences in the dataset were divided into simple, compound and complex sentences and BLEU score was computed for each of them

separately. The values are shown in Table 6.1.

Table 6.1 BLEU Scores for different types of sentences

Translation System	BLEU Score		
	Simple	Compound	Complex
Our MT System	0.588	0.557	0.498
Google Translate	0.262	0.457	0.452

It can be seen from Table 6.1 that the BLEU score is high for simple sentences and decreases further for compound and complex sentences. This decrease in BLEU score is mainly due to the inadequacy of Enconversion rules. Rules to handle phrases, clauses, certain kinds of compound sentences etc. have not been written. Inadequacy of words in the dictionary and the errors produced by Morphological Generator also contribute a small percentage to the lower BLEU score. The higher BLEU score for compound and complex sentences in contrast to simple sentences, in case of Google translate is due to the Example Based approach. The presence of commonly occurring phrases and clauses in the corpus cause the translation to be better. But the results clearly show that by adding Enconversion rules, the BLEU of our MT System can be shown to be much higher than the BLEU of Google translate for compound and complex sentences.

English-Hindi MT has been done using UNL and a BLEU score of **0.26** has been obtained[14]. They had tested their system on 60 sentences taken from agricultural corpus. When our MT system is tested with a technical corpus, it is expected that the BLEU score will reduce because of inadequacy of technical words in the dictionary and inadequacy of rules to handle more complex types of sentences. Thus by adding more Enconversion rules, a higher BLEU score can be obtained for any type of corpus. Enhancement of Morphological Generator would

also attribute to the increase the BLEU.

The Literature Survey shows that a BLEU score of 0.6950 has been obtained using Context Based Machine Translation (CBMT) approach for Spanish, Arabic, Chinese to English[3]. The characteristics of the source and target languages also play a part in the efficiency of the approach used for translation. Being a corpus based approach, this method requires huge backend corpora which makes it inefficient. Other methods to MT report a BLEU score of 0.16 - 0.20 only showing the superiority of UNL approach.

6.4.2 Fluency and Adequacy

6.4.2.1 Description

These are human evaluated scores which determine the efficiency of Machine Translation. Fluency refers to the degree to which the target is well formed according to the rules of target language grammar. A fluent segment is one that is well-formed grammatically, contains correct spellings, is intuitively acceptable and can be sensibly interpreted by a native speaker of that language. Adequacy refers to the degree to which the information present in original sentence is also communicated in translated sentence.

6.4.2.2 Evaluation and Results

To measure fluency and adequacy, the input and the translated output sentences of our system as well Google translate were given to around 60 people in the age groups 20 - 60 years. A scale of 0-5 was used for both of these measures. The description of Fluency scale is shown in Table 6.2 and the description of Adequacy scale is shown in Table 6.3.

Table 6.2 Scale to evaluate Fluency

Score	Level	Description
5	Perfect	Good grammar
4	Fair	Understandable, minor grammatical errors
3	Acceptable	Understandable, flawed grammar
3	Bad	Broken, understandable with effort
1	Poor	Not understandable
0	Nonsense	Incomplete, makes no sense

Table 6.3 Scale to evaluate Adequacy

Score	Level	Description
5	All	No loss of meaning
4	Most	Most of the meaning conveyed
3	Acceptable	Noun and Tense information conveyed
3	Some	Noun information alone conveyed with proper gender
1	Few	Noun information alone conveyed with wrong gender
0	None	No meaning conveyed

Table 6.4 gives average score of fluency and adequacy for our MT system as well as Google Translate.

Table 6.4 Results of Fluency and Adequacy tests

Translation System	Fluency Score		Adequacy Score	
	Age (20-40) yrs	Age (Above 40) yrs	Age (20-40) yrs	Age (Above 40) yrs
Our MT System	3.957	3.867	4.011	3.833
Google Translate	2.96	2.9	2.71	2.6

The number of sentences that fall under each category of these scores has been computed for both the systems and the results can be seen in figure 6.10 and 6.11.

During these tests, it was found that our system always produced the correct tense and gender of the subject. This is evident from the very

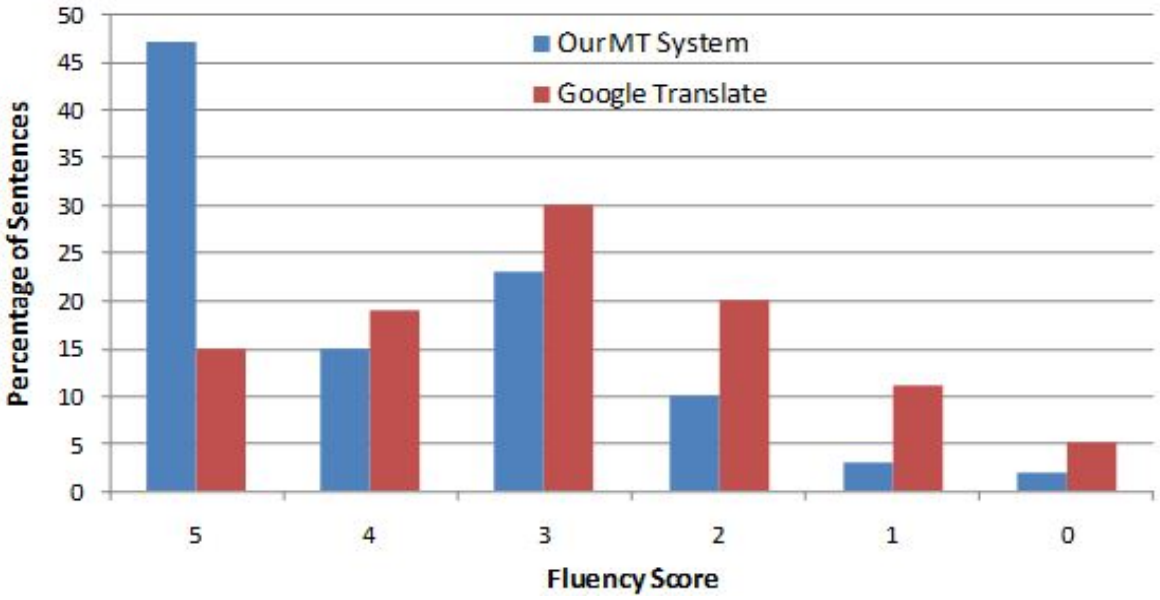

Figure 6.10 Bar chart showing the percentage of sentences under each category of score – Fluency

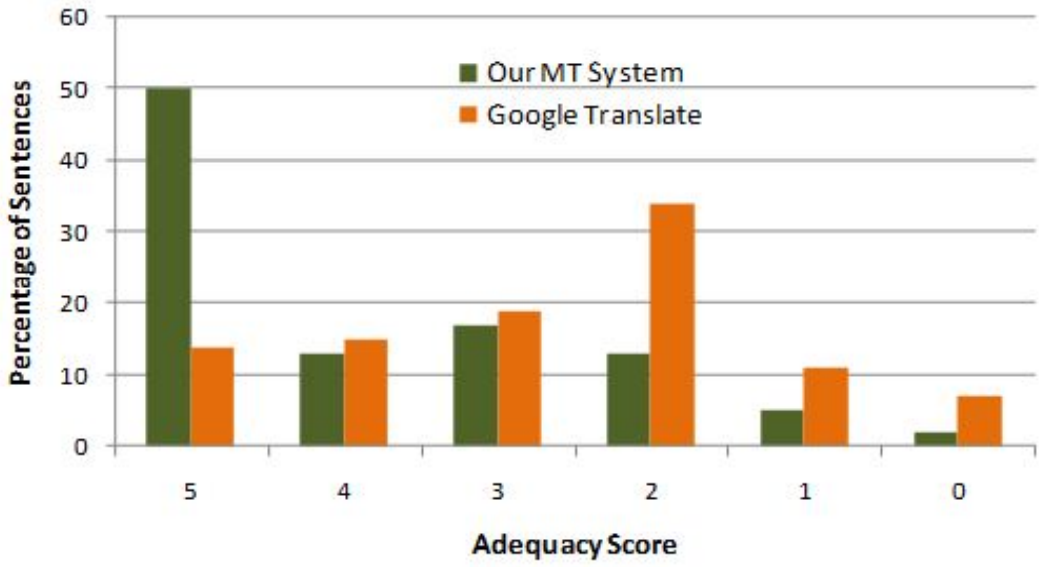

Figure 6.11 Bar chart showing the percentage of sentences under each category of score – Adequacy

few number of sentences with a score of 0,1 and 2 for adequacy. On the contrary, Google translate did not convey tense and gender information for most of the sentences as seen from figure 6.11. Moreover, the ordering of words in Google translate was inefficient which resulted in loss

of meaning in the target sentence. Figure 6.10 shows this with fluency scores being less than or equal to 3 for most of the sentences. It can be observed that the UNL approach provides a very good fluency forming grammatically correct sentences.

These results show that UNL approach helps to retain the meaning of the input sentence and helps to achieve better grammatical correctness. These high scores are also attributed to the rules of the Morphological Generator as they play a major role in the formation words of target sentence. Thus improving the Enconversion rule base and increasing the efficiency of the Morphological Generator, higher scores can be obtained.

6.4.3 Word Error Rate(WER)

WER is a common metric to evaluate the performance of a MT system. It is determined by calculating the Levenshtein distance between those words in the candidate translation and the reference translation which have a common prefix of atleast 3. The Levenshtein distance essentially gives the number of additions, deletions and modifications. WER is computed as

$$Word\,Error\,Rate\,WER = \frac{S+D+I}{N} \tag{6.1}$$

In equation 6.1,

- S is the number of substitutions
- D is the number of deletions
- I is the number of insertions
- N is the number of letters in the reference word

WER is calculated for every word in the sentences of the candidate translation. The results are then averaged over all the sentences in the corpus. WER for our system using UNL approach was calculated to be 35.6%

whereas for Google translate it was 59%. Figure 6.12 gives the WER for simple, compound and complex sentences in the dataset.

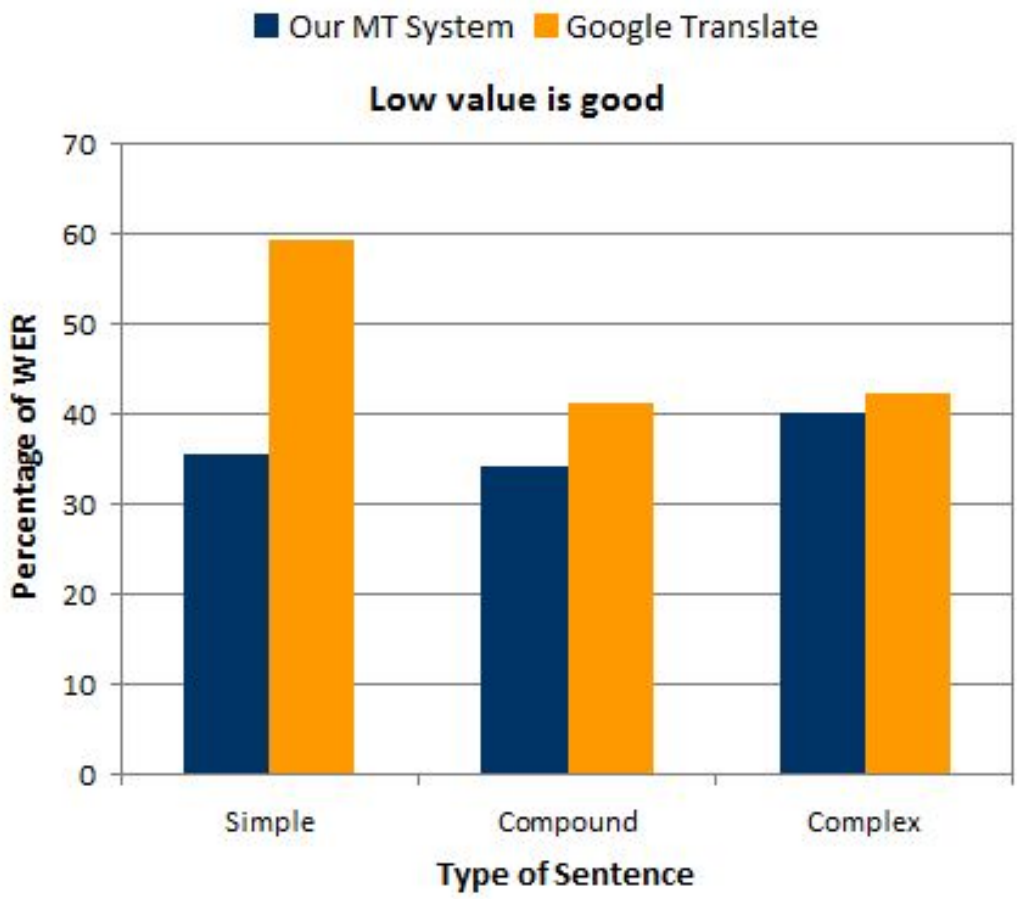

Figure 6.12 Bar chart showing the percentage of WER for different types of sentences

The data in figure 6.12 clearly shows the advantages of incorporating a Morphological Generator in the system. Google translate just uses a corpus and hence no complete sentence is formed conveying the tense and gender of the subject.

These results show that the Morphological Generator is quite efficient in that it generates words much closer to the ones in the reference translation. Adding more rules to the Morphological Generator, its efficiency can be improved thus reducing the percentage of WER.

CHAPTER 7

CONCLUSIONS

7.1 SUMMARY

This is a machine translation system which translates an English sentence to Tamil using UNL as the intermediate. UNL helps to retain the meaning of the input sentence and also to present the output sentence in a readable form by manipulating the relationships of UNL. The system uses a POS Tagger to identify the part-of-speech of every word in the sentence and a Morphological Analyser to obtain the root words. The initial UW List is created by adding attributes and domain constraints which is then given to the Enconverter. An UNL Expression is formed by the Enconverter. Tamil words for the HWs are extracted from the dictionary and then complete words are formed using the Morphological Generator. Finally the sentence formation algorithm produces a meaningful Tamil sentence.

Certain modifications were made to the standard UNL representation, to help in the process of Deconversion. One such modification involves adding information about gender, person and number in the verb node of UNL. This is essential since the construction of complete Tamil verb requires PNG information. The Enconversion algorithm was also modified to consider only two nodes at a time for simplicity. This made the process of Enconversion and writing rules simple but it required adding more new attributes to help in the formation of relationships. The results of performance evaluation are very encouraging and show

that the efficiency of UNL approach to MT is about 82%. The results of fluency and adequacy tests reveal that the Morphological Generator is 85% efficient.

7.2 CRITICISMS

The errors in the POS Tagger propagate down to all the modules of the system and influenced in corrupting the meaning of translated output. Most of the errors in the system have been found to be due to incorrect POS. A few errors have been found in the spelling of the target words and adding of suffixes to root words. These are due to errors in the Morphological Generator as this was unable to handle exceptions that are specific to Tamil language. For example, there were no specific rules in Tamil grammar to determine the tense marker for the past form of the verb. A lot of exceptions exist in every category of rules of the Morphological Generator. The performance evaluation of the system reveals the inability of the system to translate some compound sentences and most of the complex sentences. The system does not handle homonyms, sentences in which two or more verbs occur consecutively, sentences with clauses, phrases, punctuation marks (comma, question mark, exclamation mark) and sentences which contain numerals. This shows that more rules should be added to the Enconversion process.

7.3 FUTURE WORK

The efficiency of Stanford POS tagger is only around 56% for sentences. Hence it could be replaced by a better POS tagger to improve the efficiency. Parsing of UNL Knowledge Base can be added to enhance the UNL which in turn would improve the quality of translation. More rules can be written to handle phrases and clauses. This would provide better results when technical documents containing complex sentences

are translated. A technique to handle homonyms and numerals need to be identified. Thus there are a lot of improvements and extensions possible and this system offers a wide scope in the field of Machine Translation.

APPENDIX A

Test Cases For Each Module

This section provides the test cases for each of the modules of the system developed.

A.1 POS TAGGER

A.1.1 Test Pre-requisite

A grammatically correct English sentence is to be given as input.

A.1.2 Description

The set of test cases to this module covers different types of English words like gerund, words denoting possession, exclamation etc.

A.1.3 Test Cases

- **TC_ ID :** 01
  Input: Sentences with proper nouns.
  Expected Output: Proper nouns should be differentiated from other nouns and tagged NNP.
- **TC_ ID :** 02
  Input: Sentences containing (’s).
  Expected Output: Tagged as POS indicating possession.
- **TC_ ID :** 03
  Input: Interrogative sentences.
  Expected Output: ”Wh” pronouns in the beginning of the sentence tagged as WP indicating interrogation.

- **TC_ ID :** 04

  Input: Exclamatory sentences containing interjective words like Oh, Ah, Alas etc.

  Expected Output: Tagged as UH.

- **TC_ ID :** 05

  Input: Sentences containing gerund or modal verbs.

  Expected Output: Gerunds are tagged as VBG and modals as MD.

A.2 MORPHOLOGICAL ANALYSER

A.2.1 Test Pre-requisite

A grammatically correct English sentence is to be given as input.

A.2.2 Description

The set of test cases to this module covers different types of English words like gerund, proper nouns, pronouns, verbs etc.

A.2.3 Test Cases

- **TC_ ID :** 01

  Input: Sentences with proper nouns or pronouns.

  Expected Output: Proper nouns and pronouns remain unaffected.

- **TC_ ID :** 02

  Input: Sentences with verbs in any of the 12 standard forms (including 'be' form of the verbs).

  Expected Output: Root of the verb. Modals like will, shall and words like has, have, had will be retained as such.

- **TC_ ID :** 03

  Input: Sentences with plural forms of nouns.

Expected Output: Root of the noun which is mostly its singular form.

- **TC_ ID :** 04

  Input: Sentences containing prepositions or articles.

  Expected Output: Prepositions and articles do not get affected and are returned as such.

- **TC_ ID :** 05

  Input: Sentences containing a gerund.

  Expected Output: Root verb is returned.

A.3 UNL ENCONVERTER

A.3.1 Test Pre-requisite

- Root words of every word in the sentence
- POS of every word in the context of the sentence
- Universal Word (obtained from the dictionary) along with all the required attributes
- Possible relationships (including hierarchical relationships) between words
- Rules of Enconversion

A.3.2 Description

Test cases for this module should be such that it covers the entire rule base and involves all the relationships specified by UNL standards. Few sample test cases are stated below.

A.3.3 Test Cases

- **TC_ ID :** 01

  Input: Sentences which contain information about the place of the event. They normally contain words like 'at' and 'in'.

Expected Output: The UNL Expression contains 'plc' relationship between the verb and the noun(place).

- **TC_ ID :** 02

  Input: Compound sentences containing conjunctions like 'and', 'but' and 'or'.

  Expected Output: 'and' and 'or' relationship can be observed between the two main verbs of each of the two simple sentences. Contrary conditions like 'but' are indicated using attributes.

- **TC_ ID :** 03

  Input: Sentences which indicate the time of the event using words like 'after', 'before', 'during', 'when', 'while' etc.

  Expected Output: UNL representation contains 'tim' relationship to denote this concept

- **TC_ ID :** 04

  Input: Sentences depicting reason using words like 'because', 'so' etc.

  Expected Output: 'rsn' relationship is observed in the UNL representation.

- **TC_ ID :** 05

  Input: Sentences denoting possession using ('s) or using words like 'my', 'mine' etc.

  Expected Output: UNL graph contains 'pos' relationship.

- **TC_ ID :** 06

  Input: Sentences containing an article followed by a noun

  Expected Output: Merged into a single Universal Word with the attributes of article added to noun.

- **TC_ ID :** 07

  Input: Sentences containing an auxiliary followed by a verb, or verbs in perfect tense that contain has/had/have.

Expected Output: Merged into a single Universal Word with root verb alone and all other information added as attributes.

A.4 TAMIL WORD EXTRACTOR

A.4.1 Test Pre-requisite

Fully formed UNL Expression in the syntax given below.

[UNL]

$<Relationship_1>(UW_1, UW_2)$

.

.

$<Relationship_n>(UW_i, UW_j)$

[/UNL]

A.4.2 Description

The test cases for this module cover all possible cases which involve extraction of the root word from the dictionary and also cases where transliteration might be required.

A.4.3 Test Cases

- **TC_ ID :** 01

  Input: UNL Expressions that contain a proper noun as a Universal Word.

  Expected Output: Modified UNL with proper noun transliterated.

- **TC_ ID :** 02

  Input: UNL Expressions containing words not found in the dictionary.

  Expected Output: Modified UNL with words not found in the dictionary transliterated by default.

- **TC_ ID :** 03

  Input: UNL Expressions containing the exact Universal Word (word + domain constraint) as in dictionary.

  Expected Output: Modified UNL with appropriate tamil word from dictionary

- **TC_ ID :** 04

  Input: UNL Expressions containing the same word as in dictionary but not the same domain constraint

  Expected Output: Modified UNL with the word being transliterated

A.5 MORPHOLOGICAL GENERATOR

A.5.1 Test Pre-requisite

- Appropriate Tamil words (root only) of each Universal Word in the UNL Expression
- Attribute information of each Universal Word

A.5.2 Description

The set of test cases to this module includes the various suffixes that might be added to a root word to obtain complete Tamil word.

A.5.3 Test Cases

- **TC_ ID :** 01

  Input: Universal Words that contain nouns in singular or plural form.

  Expected Output: Number suffix and Case suffix are added based on the attributes to form the complete noun.

- **TC_ ID :** 02

  Input: Universal Words that contain verbs in one of the 12 stan-

dard tense forms.

Expected Output: Suffixes are added to verbs to indicate tense according to Tamil grammar rules.

A.6 SENTENCE FORMATION

A.6.1 Test Pre-requisite

- Completely formed Tamil words
- Rules of sentence formation (Tamil grammar rules)

A.6.2 Description

The test cases to this module depicts the various cases where re-ordering of words occur during the construction of the translated output.

A.6.3 Test Cases

- **TC_ ID :** 01

  Input: Tamil words of sentences containing only subject, verb and object

  Expected Output: Output Tamil sentence rearranged in the order subject followed by object and then verb.

- **TC_ ID :** 02

  Input: Tamil words of compound sentences which contain words like 'but', 'after', 'while' or relationships like 'rsn', 'tim'

  Expected Output: Output Tamil sentence in which the latter simple sentence comes first and the former follows it.

APPENDIX B

Rules Used In The System

This section gives a subset of the rules used in the process of Enconversion and Deconversion in our MT System.

B.1 ENCONVERSION RULES

This section gives a set of sample rules used in the Enconversion process. The rules given here follow the template described in Section 4.4.5(Enconversion Rule Base)

B.1.1 Rules used for merging nodes

These are rules which merge two nodes and add required attributes of one node to another. These rules determine the final list of UWs that would appear in the output sentence.

1. (not,,),(,,.@verb):=(0,,,,),(2,.@NEG,,,)
2. (do,,),(,,.@verb):=(0,,,,),(2,,,%1.TENSE,)
3. (have,,.@NEG),(,,.@verb):=(0,,,,),(2,.@NEG,,%1.TENSE,)
4. (have,,),(,,.@verb):=(0,,,,),(2,,,%1.TENSE,)
5. (be,,.@verb.@NEG),(,,.@verb.@progressive):=(0,,,,),
   (2,.@NEG,.@present,%1.TENSE,)
6. (be,,.@verb),(,,.@verb.@progressive):=(0,,,,),
   (2,,.@present,%1.TENSE,)
7. (,,.@def),(,,.@adjective):=(0,,,,),(2,.@DEF,,,)
8. (,,.@indef),(,,.@adjective):=(0,,,,),(2,.@INDEF,,,)
9. (,,.@def.@adjective),(,,):=(2,,.@def,,),(2,.@def,,,)

10. (,,.@indef.@adjective),(,,):=(2,,.@indef,,),(2,.@indef,,,)
11. (,,.@def),(,,.@noun):=(0,,,,),(2,.@DEF,,,)
12. (,,.@indef),(,,.@noun):=(0,,,,),(2,.@INDEF,,,)
13. (,,.@aux),(be,,):=(0,,,,),(2,.@future,,,)
14. (,,.@aux),(,,.@verb):=(0,,,,),(2,.@future,.@present,,)
15. (,,.@perfect.@NEG),(,,.@verb.@progressive):=(0,,,,),
16. (2,.@NEG,.@present,%1.TENSE,)
17. (,,.@perfect),(,,.@verb.@progressive):=(0,,,,),
    (2,,.@present,%1.TENSE,)
18. (to,,),(,,.@noun):=(0,,,,),(2,.@TO,,,)
19. (to,,),(,,.@pronoun):=(0,,,,),(2,.@TO,,,)
20. (via,,),(,,.@noun):=(0,,,,),(2,.@VIA,,,)
21. (through,,),(,,.@noun):=(0,,,,),(2,.@VIA,,,)
22. (from,,),(,,.@noun):=(0,,,,),(2,.@FROM,,,)
23. (from,,),(,,.@pronoun):=(0,,,,),(2,.@FROM,,,)
24. (using,,),(,,.@noun):=(0,,,,),(2,.@USING,,,)

B.1.2 Rules used to form UNL relationships

These are some rules which are used to form relationships between any two UWs. These relationships depict the semantic relation between the words of a sentence. Therefore they help to convey the meaning of the input sentence in the translated sentence.

1. (,,.@verb),(,,.@VIA):=(1,via,%1,%2,),(0,,,,)
2. (,,.@noun),(,,.@VIA):=(1,via,%1,%2,),(0,,,,)
3. (,,.@verb),(,,.@FROM):=(1,frm,%1,%2,),(0,,,,)
4. (,,.@noun),(,,.@FROM):=(1,frm,%1,%2,),(0,,,,)
5. (some,,),(,,.@noun):=(0,,,,),(1,qua,%1,%2,)
6. (many,,),(,,.@noun):=(0,,,,),(1,qua,%1,%2,)
7. (,,.@noun),(,,.@AND.@noun):=(0,,,,),(1,and,%1,%2,)

8. (,,.@verb),(,,.@adverb):=(1,aoj,%2,%1,),(0,,,,)
9. (at,,),(,,.@noun):=(0,,,,),(1,plc,)
10. (,,.@noun),(,,.@verb):=(0,,,,),(1,agt,%2,%1,)
11. (,,.@verb),(,,.@noun):=(1,obj,%1,%2,),(0,,,,)
12. (,,.@verb),(,,.@pronoun):=(1,obj,%1,%2,),(0,,,,)

B.2 MORPHOLOGICAL GENERATOR RULES

Following are some of the rules which are part of our Morphological Generator. It consists of rules to determine the tense marker and PNG suffix for Tamil verbs and case suffix for Tamil nouns.

- Tamil verbs generally take the form Root Verb + Tense Marker + PNG Suffix
- Tamil Nouns generally take the form Noun + Case Suffix (optional)

To obtain perfect and continuous tense forms of the verb, the participle form of the verb needs to be generated for which separate rules are written.

B.2.1 Rules to determine PNG suffix

Figure B.1 shows the various suffixes that will be added to the root Tamil verb to form the complete Tamil verbs based on the person, number and gender information. It can be seen that in case of first and second person, the suffix to be added remains the same, irrespective of the gender whereas in third person, the suffix varies depending on the gender being male, female, common or neuter.

B.2.2 Rules to determine tense marker

Figure B.2 gives the rules to determine the tense marker for future tense. Similarly rules are written to determine the tense marker for present and past tense forms of the verb. Rules are also written to form

PERSON	GENDER	NUMBER	SUFFIX
FIRST		SINGULAR	ஏன்
		PLURAL	ஓம்
SECOND		SINGULAR	ஈர்
		PLURAL	ஈர்கள்
THIRD	MASCULINE	SINGULAR	ஆன்
		PLURAL	ஆர்கள்
	FEMININE	SINGULAR	ஆள்
		PLURAL	ஆர்கள்
	COMMON	SINGULAR	ஆர்
		PLURAL	ஆர்கள்
	NEUTER	SINGULAR	அது
		PLURAL	அன

Figure B.1 PNG Suffix to be added to verbs

Future tense in Tamil has two tense markers namely ப் and வ்.

- If it is a strong verb, which means it satisfies வல்லினம் மிகும் rules if the நிலைமொழி முதல் is a வல்லினம், then the tense marker is ப்
  - Eg: படி - படிப்பேன்
- If it is a weak verb, which means it does not satisfy வல்லினம் மிகும் rules if the நிலைமொழி முதல் is a வல்லினம், then the tense marker is வ்.
  - Eg: பாடு - பாடுவேன்

Figure B.2 Rules to determine Tense Marker - Future Tense

the perfect and continuous tense forms of the Tamil verbs by generating their participle. Apart from all these information, rules are also required to be written to determine how the root Tamil words combine with the suffixes to form the complete Tamil word.

B.2.3 Rules to determine case suffix

The case suffix to be added to the nouns is identified from the UNL relationships in which they are involved. Figure B.3 gives a mapping of a few UNL relations to the suffixes to be added. These suffixes are added to the nouns after generating their plural forms if necessary.

RELATION	SUFFIX
obj	ஐ
to	க்கு
via	வழியாக
pof	இன்
ben	க்காக
cag	உடன்
dur	போது
met	இல்
frm	இலிருந்து
qua	சில, பல
aoj	ஆக
and	மற்றும்
or	அல்லது

Figure B.3 Mapping of Case Suffix to UNL Relations

REFERENCES

[1] MBA Salai Aaviyamma and K. Kathiravan, "Problems related to Eng-Tam Translation", In *Proceedings of the International Forum for Information Technology in Tamil*, pp. 169–172, 2009.

[2] Y. Choueka Bar, Kfir and N. Dershowitz, "An Arabic to English example - based translation system", In *Proceedings of Information and Communication Technologies International Symposium and Workshop on Arabic Natural Language Processing*, pp. 355–359, 2007.

[3] Jaime G Carbonell, Steve Klein, David Miller, Mike Steinbaum, Tomer Grassiany, and Jochen Frey, "Context-based machine translation", *The Association for Machine Translation in the Americas*, pp. 19–28, 2006.

[4] Michael Carl, Cathrine Pease, Leonid L Iomdin, and Oliver Streiter, "Towards a dynamic linkage of example-based and rule-based machine translation", *Machine Translation*, vol. 15, num. 3, pp. 223–257, 2000.

[5] Michael Carl, Andy Way, and Walter Daelemans, "Recent advances in example-based machine translation", *Computational Linguistics*, vol. 30, num. 4, pp. 516–520, 2004.

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

[7] Virach Sornlertlamvanich Charoenpornsawat, Paisarn and Thatsanee Charoenporn, "Improving translation quality of rule-based machine translation", In *Proceedings of the Association for Computational Linguistics COLING workshop on Machine translation in Asia*, volume 16, pp. 1–6, 2002.

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

[9] Jignashu Parikh Dave, Shachi and Pushpak Bhattacharyya, "Interlingua-based EnglishHindi Machine Translation and Language Divergence", *Machine Translation*, pp. 251–304, 2001.

[10] T. Dhanabalan and T. V. Geetha, "UNL Deconverter for Tamil", In *International Conference on the Convergences of Knowledge, Culture, Language and Information Technologies*, 2003.

[11] Gregory Grefenstette, "The World Wide Web as a resource for example-based machine translation tasks", In *Proceedings of the ASLIB Conference on Translating and the Computer*, volume 21, 1999.

[12] Tarcisio Della Senta Hiroshi Uchida, Meiying Zhu, *Universal Networking Language*, UNDL Foundation, Tokyo, Japan, 2005.

[13] Margaret King Hovy, Eduard and Andrei Popescu-Belis, "Principles of context-based machine translation evaluation", *Machine Translation*, vol. 17, num. 1, pp. 43–75, 2002.

[14] Manoj Jain and Om P. Damani, "English to UNL (Interlingua) Enconversion", In *Proceedings of the 2nd Conference on Language and Technology*, pp. 1–8, 2009.

[15] Daniel Jones, "Non-hybrid example-based machine translation architectures", In *Proceedings of the 5th International Conference on Theoretical and Methodological Issues in Machine Translation*, pp. 163–171, 1992.

[16] Yves Lepage and Etienne Denoual, "Purest ever example-based machine translation: Detailed presentation and assessment", *Machine Translation*, vol. 19, num. 3, pp. 251–282, 2005.

[17] Amitabha Mukerjee, Achla M Raina, Kumar Kapil, Pankaj Goyal, and Pushpraj Shukla, "Universal Networking Language: A Tool for Language Independent Semantics?", *Universal Networking Language: Advances in Theory and Applications*, pp. 145–150, 2003.

[18] ThuyLinh Nguyen and Stephan Vogel, "Context-based Arabic morphological analysis for machine translation", In *Proceedings of the Association for Computational Linguistics Twelfth Conference on Computational Natural Language Learning*, pp. 135–142, 2008.

[19] Constantine Domashnev Nirenburg, Sergei and Dean J. Grannes, "Two approaches to matching in example-based machine translation", In *Proceedings of the 5th International Conference on Theoretical and Methodological Issues in Machine Translation*, pp. 47–57, 1993.

[20] Sergei Nirenburg, "Knowledge-based machine translation", *Machine Translation*, vol. 4, num. 1, pp. 5–24, 1989.

[21] Bhattacharyya P. Hegde J. Shah R. M. Ramanathan, A. and M. Sasikumar, "Simple syntactic and morphological processing can help English-Hindi statistical machine translation", In *Proceedings of International Joint Conference on Natural Language Processing*, pp. 513–520, 2008.

[22] S Saraswathi, M Anusiya, P Kanivadhana, and S Sathiya, "Bilingual Translation System for Weather Report", In *Proceedings of the International Conference on Advances in Computing and Communications*, pp. 155–164, 2011.

[23] R. M. K Sinha and A. Jain, "AnglaHindi: an English to Hindi machine-aided translation system", In *Proceedings of MT Summit IX*, pp. 494–497, New Orleans, USA, 2003.

[24] Harold Somers, "Review article: Example-based machine translation", *Machine Translation*, vol. 14, num. 2, pp. 113–157, 1999.

[25] Shrikanth Narayanan Sridhar, Vivek Kumar Rangarajan and Srinivas Bangalore, "Enriching spoken language translation with dialog acts", In *Proceedings of Association for Computational Linguistics, Short Papers (Companion Volume)*, pp. 225–228, 2008.

[26] Sohail Asghar Tahir, Ghulam Rasool and Nayyer Masood, "Knowledge Based Machine Translation", In *Proceedings of the IEEE International Conference on Information and Emerging Technologies*, pp. 1–5, 2010.

[27] D. Thenmozhi and C. Aravindan, "Tamil-English Cross Lingual Information Retrieval System for Agriculture Society", In *Proceedings of the International Forum for Information Technology in Tamil*, pp. 173–178, 2009.

[28] Konstantin Tretyakov, "Example-Based Machine Translation of Short Phrases Using the Context Equivalence Principle", 2007.

[29] M Tynovsky, "Hybrid Approaches in Machine Translation", In *WDS Proceedings of Contributed Papers, Part-I*, pp. 124–128, 2008.

[30] Ruvan Weerasinghe, “A Statistical Machine Translation Approach to Sinhala-Tamil Language Translation”, *Towards an ICT enabled Society*, pp. 136–141, 2003.

[31] Hua Wu and Haifeng Wang, “Pivot language approach for phrase-based statistical machine translation”, *Machine Translation*, vol. 21, num. 3, pp. 165–181, 2007.

[32] Franz Och Zens, Richard and Hermann Ney, “Phrase-based statistical machine translation”, In *Proceedings of KI 2002: Advances in Artificial Intelligence*, pp. 35–36, 2002.

[33] Richard Zens and Hermann Ney, “Improvements in phrase-based statistical machine translation”, In *Proceedings of Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 257–264, 2004.

[34] Andreas Zollmann and Ashish Venugopal, “Syntax augmented machine translation via chart parsing”, In *Proceedings of the Association for Computational Linguistics Workshop on Statistical Machine Translation*, pp. 138–141, 2006.