*Sādhanā* Vol. 41, No. 6, June 2016, pp. 607–620
DOI 10.1007/s12046-016-0504-9

# English to Tamil machine translation system using universal networking language

RAJESWARI SRIDHAR, PAVITHRA SETHURAMAN* and KASHYAP KRISHNAKUMAR

College of Engineering Guindy, Anna University, Guindy, Chennai 600 025, India
e-mail: itzmepavithu@gmail.com

MS received 20 June 2014; revised 29 February 2016; accepted 1 March 2016

**Abstract.** This paper proposes English to Tamil machine translation system, using the universal networking language (UNL) as the intermediate representation. The UNL approach is a hybrid approach of the rule and knowledge-based approaches to machine translation. UNL is a declarative formal language, specifically designed to represent semantic data extracted from a natural language text. The input English sentence is converted to UNL (enconversion), which is then converted to a Tamil sentence (deconversion) by ensuring that the meaning of the input sentence is preserved. The representation of UNL was modified to suit the translation process. A new sentence formation algorithm was also proposed to rearrange the translated Tamil words to sentences. The translation system was evaluated using bilingual evaluation understudy (BLEU) score. A BLEU score of 0.581 was achieved, which is an indication that most of the information in the input sentence is retained in the translated sentence. The scores obtained using the UNL based approach were compared with existing approaches to translation, and it can be concluded that the UNL is a more suited approach to machine translation.

**Keywords.** Design; algorithms; performance; languages; machine translation; UNL; enconverter; deconverter; morphological generator; POS tagger; hybrid approach.

## 1. Introduction

Natural language processing is a field of Computer Science, which deals with interactions between the computer and human languages [1]. The interaction is essential in order to make the computer understand and interpret human language. Computational linguistics is the term used in natural language processing that discusses the modelling of a natural language from a computational perspective. Machine translation (MT) is a sub-field of computational linguistics that investigates the use of software to translate a text or speech from one natural language to another [2]. In this paper, we discuss the machine translation of a text from English to Tamil.

Text translation from one language to another is usually done using a dictionary (source to target language), which is a tedious job. A dictionary will be able to do a word-by-word translation, but in order to deliver the complete meaning of the input sentence and its context, translation at the sentence level is essential. As the process of sentence level translation is tedious for human beings, automating this process could help in many applications, like translation of books, legal documents, press reports of the media, and any other piece of text to aid better understanding. For a machine to do sentence level translation, in addition to the dictionary, we need to provide the grammar of the source and target languages. After providing the grammar and the dictionary, other issues like word sense disambiguation, preserving the meaning of the sentence, grammatical correctness of the translated sentence, handling various dialogue acts, and the differences between the source and target languages, need to be considered. Researchers have been working on automated translation between languages for nearly two decades [3].

In this paper, we discuss a machine translation (MT) system that would translate an English sentence into Tamil, using the 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 the people of South India and North-East Sri Lanka. Tamil is the one of the few oldest languages, and its grammar is dated before 4 B.C. [http://en.wikipedia.org/wiki/Tolkappiyam]. Thus, considering the importance of these two languages, we chose to develop this MT system.

This paper is organized as follows: section 2 gives a brief literature survey of the various approaches available to MT. Section 3 describes the design of our proposed approach and section 4 explains the design in detail. Section 5 outlines the results of this approach to MT. This section describes the various parameters used to test the system,

*For correspondence

and the results obtained. Finally, section 6 concludes the paper, with possible future work.

## 2. Literature survey

Many approaches have been described for MT, some of which are shown in figure 1.

Phrase based translation (PBT) involves identifying phrases in the source language text, and translating them into the 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 PBT [4]. Here, phrases can even be a group of words (substring), not necessarily syntactic phrases. A simple PBT system was discussed by Zens *et al* [5].

Many additional features, such as the addition of the dialogue act tagger [6], and incorporation of syntactic and morphological information [7] have been added to this simple system, to improve the translation quality. Certain other improvements in PBT have been suggested by Zens and Ney [8]. A new approach to PBT using a pivot language (intermediate language) has been proposed by Wu and Wang [9]. PBT has been used to translate Japanese, Chinese and Mandarin languages to English.

Example based MT (EBMT) is aimed at emulating the way humans learn and do translation. 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, Grefenstette used this as a resource for EBMT [10]. A typical example based approach to machine translation involves matching, transfer and recombination. Matching refers to searching the corpus for fragments of source text. Extracting the corresponding fragment in target language is called transfer and the process of combining together all the translated fragments and finding 'N' best re-combinations that cover the sentence is called recombination. Researchers have been proposing various techniques for matching in EBMT. Two approaches to matching, other than the canonical ones, are discussed by Domashnev *et al* [11]. This approach was used for Arabic to English, and a BLEU score of 0.1849 was obtained [12]. EBMT has also been discussed by Somers [13].

Applying EBMT to short phrases using the context equivalence principle, which avoids word-to-word alignment, has been proposed by Tretyakov [14]. A non-hybrid EBMT is a pure EBMT which involves proportional analogy, whereas a hybrid EBMT involves modifications and additions such as paraphrases to a pure EBMT. The difference between the hybrid and pure EBMT has been described by Jones [15], who adopted the non-hybrid EBMT over hybrid EBMT. Lepage *et al* [16] have also explained a pure EBMT based approach to translation. 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 Carl *et al* [17].

Rule based machine translation (RBMT) 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 statistical machine translation (SMT). Chinese to English translation has been done using this method [18]. Another approach to RBMT using chart parsing has been proposed by Zollmann *et al* [19]. A basic SMT approach was tried for Sinhala to Tamil translation [20]. Aaviyamma and Kathiravan [21] have described the process of translation to consist of four steps,

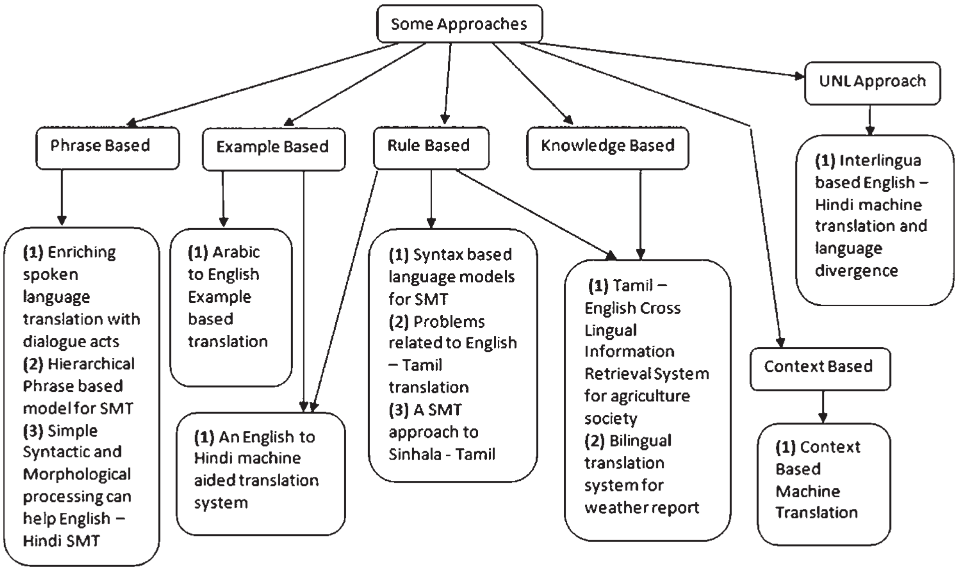

Some Approaches
Phrase Based
Example Based
Rule Based
Knowledge Based
UNL Approach
Context Based
(1) Enriching spoken language translation with dialogue acts
(2) Hierarchical Phrase based model for SMT
(3) Simple Syntactic and Morphological processing can help English – Hindi SMT
(1) Arabic to English Example based translation
(1) An English to Hindi machine aided translation system
(1) Syntax based language models for SMT
(2) Problems related to English – Tamil translation
(3) A SMT approach to Sinhala - Tamil
(1) Tamil – English Cross Lingual Information Retrieval System for agriculture society
(2) Bilingual translation system for weather report
(1) Interlingua based English – Hindi machine translation and language divergence
(1) Context Based Machine Translation

**Figure 1.** Broad classification of the approaches to machine translation.

namely, tokenisation, parsing, word mapping and sentence structure changing to translate English to Tamil. Some techniques which help in word sense disambiguation in RBMT by using contextual information have been suggested by Paisarn Charoenpornsawat *et al* to improve the correctness of translation [22].

The basic principle behind context based machine translation (CBMT) is to create an 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 for this purpose. An approach to evaluate the CBMT is discussed by King *et al* [23]. The CBMT is close to the example based approach mentioned earlier, and does not require a parallel corpus. It is suitable only for those languages which are more context dependent than rule dependent [24]. Nguyen and Vogel [25] have given a more detailed insight into the CBMT technique, for the morphological analysis of Arabic. CBMT has been used for translating Spanish to English, Arabic to English and Chinese to English languages.

Tynovsky [26] proposed a combination of approaches, known as the hybrid model by combining more than one approach to perform machine translation. A combination of rule and example based approaches was proposed by Sinha and Jain to translate from English to Hindi [27]. Here, the 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 were not found in the corpora being used, rule-based translation was done. An approach to translation involving this hybrid system has also been explained by Carl *et al* [28]. A combination of rule and knowledge based approaches helps to improve rule based translation by incorporating learning [29]. Hybrid approach to machine translation serves as the basis for cross lingual information systems. The bilingual translation system for weather report [30] has been developed using this approach that translates from English to Tamil. Tahir *et al* [31] have proposed English to Urdu translation, using knowledge based machine translation (KBMT). The KBMT-89 project at Carnegie Mellon University's "Centre for Machine Translation" is discussed by Nirenburg [32]. This system translates between the English and Japanese languages.

All these approaches to machine translation which have been discussed so far, looked at the source and target languages specifically and do not support a chain of language translations. In order to support translation between any two languages an intermediate representation is necessary. This is the basis for the universal networking language developed by UNDL Foundation [http://www.undl.org]. Universal networking language (UNL) is a declarative formal language specifically designed to represent semantic data extracted from natural language texts [33].

A UNL is composed of universal words (UW), attributes and relations.

- A UW represents simple or compound concepts. It is made up of a character string called the head word (HW) followed by a constraint list given by *<UW>:= <HeadWord> [<Constraint List>].* Example: spring (icl > tool), spring (icl > season). The HW in the UNL representation is often an English word equivalent to the source language word.
- The constraint list restricts the interpretation of a UW to a subset or to a specific concept. In the above example "(icl > tool)'' is the constraint list where "icl" stands for 'is a kind of'.
- The constraint list is followed by a list of attributes, which provide information about how the concept is being used in a particular sentence. These attributes may denote time (@past, @present), speaker's focus (@entry, @emphasis), speaker's attitudes (@exclamation, @interrogative), etc.
- UNL relations are binary and are formed between two UWs. For example: "agt" relationship defines a thing that initiates an action – agt(do, thing) where 'thing' initiates the action 'do'. Example, consider the sentence "Ram walked" – agt(walk, Ram).
- Consider the sentence: John is reading a novel. The corresponding UNL representation would be

[UNL]
agt(read(icl>do) @entry.@present.@progress, John(iof>person))
obj(read(icl>do) @entry.@present.@progress, novel(icl>book))
[/UNL]

Hence, if an input sentence of a source language is converted to the UNL representation (enconversion), then the system needs to process this UNL representation and can convert to any target language by adopting the rules of the target language (deconversion). UNL has been designed for supporting machine translation and hence, UNDL justifies it as the best approach towards designing language independent systems [http://www.undl.org]. Having enconverters and deconverters for multiple languages, we could perform translation between two languages, just by modifying the UNL obtained from the enconversion of source language, to match the UNL that would be constructed from the enconversion of target language, so that this modified UNL would aid the deconversion to target language. 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 by incorporating rules in the enconversion and deconversion algorithms. Thus, there is a lot of scope in using UNL for machine translation. Mukherjee *et al* indicate that the UNL structure can be used to derive a lot of inferences for machine learning applications [34]. Using this approach, an English–Hindi translation system has been designed [35], and a sentence level translation rate of 95% was obtained. The

UNL based approach can be thought of as a hybrid of the rule and knowledge based approaches. Researchers have written enconverters and deconverters for various Indian languages like Malayalam [36], Punjabi [37], Hindi [35] and Bangla [38]. However, with regard to Indian languages, end to end translation using UNL has been attempted only for English to Hindi [35, 39].

After carefully considering and analyzing the various approaches to machine translation, we considered to work on English to Tamil translation system for reasons already discussed. Since Tamil is a free word order language, no individual approach mentioned in figure 1 will address the language issues and hence we need to go in for a hybrid approach. Moreover, English sentences follow the subject, verb, object (SVO) pattern and Tamil sentences mostly follow the SOV pattern [29]. Hence, it is necessary to incorporate the rules of the language in the translation so that a grammatically correct target sentence is obtained. It was also inferred that incorporating syntactic and morphological information could result in better 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, a knowledge base is essential. Hence, we decided to proceed with a hybrid approach of rule and knowledge based systems, viz, the UNL based approach.

We also wanted to move towards designing a language independent 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. In this work, we implemented an enconvertor for English to UNL, and a deconvertor for UNL to Tamil, in addition to incorporating the other modules that are necessary for translation. The work of Dave and Bhattacharyya [35] cannot be adopted as is, since 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, where the free word order is between words of a sentence, and the sentences normally follow the subject, object, verb (SOV) pattern, but there is no guarantee that they always do so [40]. This makes it clear, that the rules would differ for the deconverters of Hindi and Tamil, and so the system developed for English to Hindi translation [35] requires changes in all modules, to adopt it for the English to Tamil translation.

Several works on enconversion have been carried out for various languages like Tamil and English [41] [35]. For Agglutinative languages like Tamil, enconversion is done by looking at the morphed word and identifying the root word. In one of the work [41], the authors have looked at the morphed suffix of the root words and have also used the neighbouring words to construct the UNL from input Tamil sentences. Since, we have done English enconversion, the fixed word order and grammar of the English language was used to construct the UNL representation for English sentences.

The following section discusses our UNL based approach to translate English sentences into Tamil.

## 3. System design

The block diagram of the UNL based system for translation from English to Tamil is shown in figure 2.

- The morphological analyser has been implemented, using JAWS API [http://lyle.smu.edu/∼tspell/jaws/index.html].
- The UNL knowledge base, UNL enconverter and the enconversion rule base have been developed as new modules in this work. 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 the UNL.
- The enconversion process has been modified to help ease the process of deconversion.
- A morphological generator for Tamil has been developed by incorporating all the possible rules of Tamil grammar. The readability of the output sentence shows the correctness of the morphological generator we have developed. We have handled all the four forms of sentences (simple, perfect, continuous, and perfect continuous) in the three tenses (present, past and

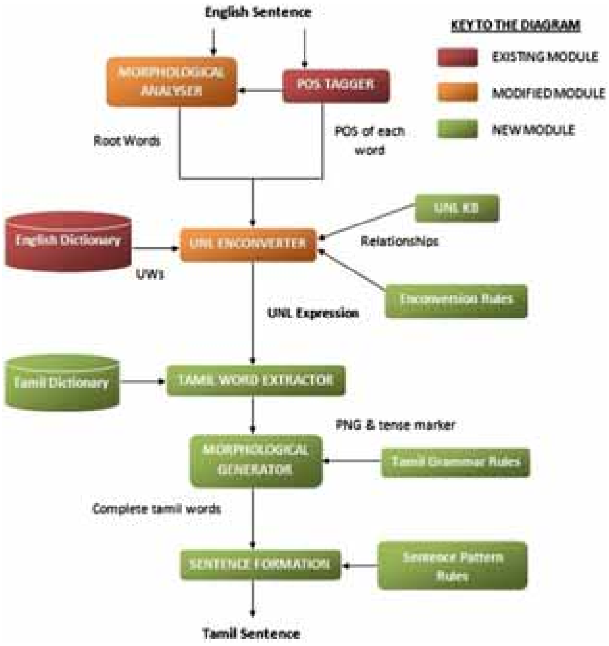


**Figure 2.** System architecture.

future). This system shows the efficiency and simplicity of using UNL for translation, thus making a contribution to the application of the UNL. It is evident from the results that this system is easily scalable.
- In addition, a new algorithm for the sentence formation of Tamil sentences has been devised, using the morphologically marked words and the grammar of Tamil language.

The system aims at translating a given English sentence to a Tamil sentence, which conveys the meaning of the input, and ensures a grammatically correct sentence as the output. The UNL wordnet dictionary developed by Ronaldo Martins was used [http://www.undl.org]. Tamil words were manually inserted into the dictionary making it a language-universal word (L-UW) 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 the root word. Domain constraints and the required attributes are added to the root word forming the initial universal word (UW) list.

This UW list is processed by the enconversion algorithm. The rules for enconversion are written following a standard template, in accordance with 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 modified dictionary. These root Tamil words are given to the morphological generator, which forms the complete Tamil words by adding the required suffixes, like the tense marker, Person Noun Gender (PNG) marker and case suffix marker, according to our newly created rules, based on Tamil grammar. Finally, the new rule-based sentence formation algorithm reorders the words, to obtain the output Tamil sentence.

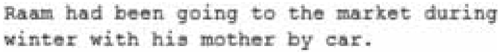
## 4. Detailed design

The discussion of the overall block diagram, and the contribution of this work in the previous section, is followed by the sub-sections that discuss each module in detail.

### 4.1 *Dictionary*

The UNL wordnet dictionary developed by Dr. Ronaldo Martins, Mackenzie University, Brazil is used [http://www.undl.org]. This dictionary consists of about 100,000 English words. This has been modified into an L-UW dictionary, by inserting Tamil words for UWs. Gender information for some specific words is also inserted. Since a single word may occur with different parts-of-speech in different contexts, multiple entries would be available for a single word in the dictionary. Thus the database is huge, and hence, querying it is time consuming. Therefore to increase the speed of retrieving words from the dictionary, we have created multicolumn indexes in the DB.

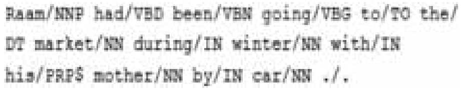
Let us consider the following example sentence and each of the following sub-sections shows the output for this sentence.



```
Raam had been going to the market during
winter with his mother by car.
```

### 4.2 *POS tagger*

The Stanford POS tagger API has been used in developing this module [http://nlp.stanford.edu/software/tagger.shtml]. The entire input sentence is given to this module, which returns the part of speech of every word in the sentence. The Stanford POS tagger identifies around 36 different parts-of-speech. However, word sense disambiguation is not carried out separately and we have considered the POS as returned by the POS tagger tool. This module is very crucial to the development of the system, and the errors that occur here penetrate through every other module following it. The following is the output of the POS tagger tool.



```
Raam/NNP had/VBD been/VBN going/VBG to/TO the/
DT market/NN during/IN winter/NN with/IN
his/PRP$ mother/NN by/IN car/NN ./.
```

In the above output, NN stands for noun and VB for verb. Variations of the verb have also been indicated.

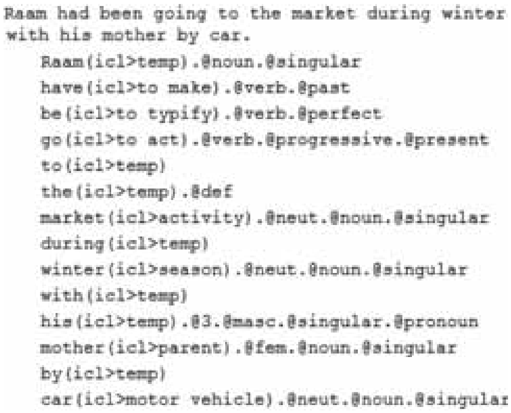
### 4.3 *Morphological analyzer*

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, in our work, we choose the first word that occurs in the dictionary as the root word. This morphological root forms the head word of that universal word (UW) in the UNL representation. In addition, the morphological analyser also forms the initial UW list from the words in the input sentence. The 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 the not null constraint. Attributes are added to the UW as and when required, until the process of enconversion is complete. The following are some of the scenarios for which attributes are added.

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

- To classify the articles (a, an, the and some) and determiners

The PNG information is required in verbs, in order to generate the suffix for the formation of the complete Tamil verb. Thus, there is a modification from the standard representation of the UNL, as stated by the UNDL. This modification to the existing UNL is essential for a partial free word order language like Tamil. Based on this modification to the UNL, the sentence generation algorithm was time efficient, which is discussed in the following sections. The following is the output of this stage for the sample input considered.



```
Raam had been going to the market during winter
with his mother by car.
    Raam(icl>temp).@noun.@singular
    have(icl>to make).@verb.@past
    be(icl>to typify).@verb.@perfect
    go(icl>to act).@verb.@progressive.@present
    to(icl>temp)
    the(icl>temp).@def
    market(icl>activity).@neut.@noun.@singular
    during(icl>temp)
    winter(icl>season).@neut.@noun.@singular
    with(icl>temp)
    his(icl>temp).@3.@masc.@singular.@pronoun
    mother(icl>parent).@fem.@noun.@singular
    by(icl>temp)
    car(icl>motor vehicle).@neut.@noun.@singular
```

### 4.4 *UNL knowledge base (KB)*

The UNL KB has been created by parsing the dictionary, and contains only the inheritance relationships between the words. 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. For example, the UNL KB contains relationships like icl(fruit, apple) and icl(human, woman) which means apple is inherited from fruit and woman is inherited from human.

### 4.5 *Enconversion rule base*

These rules aid the process of enconversion. In this work, we have considered the rules as two sets, so as to help during deconversion. The two sets of rules are

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

Rules have been written using a common template to aid their parsing. In this work, we have considered compound sentences as two separate simple sentences, and hence, rules are written only to create the compound relationship. Rules are written to handle all the 46 relationships of the UNL as described by the universal networking digital language (UNDL) [33].

Some examples of UNL enconversion rules are given below.

(1) (,,.@def),(,,.@noun):=(0,,,,),(2,.@DEF,,,) – This rule indicates that the first UW should have attribute @def(definite) and the second one should have the attribute @noun. When these conditions are satisfied, the first UW gets deleted, and the second one undergoes a modification, i.e., the addition of an attribute @DEF. This rule is used to combine the definite article "the" and a noun following it. For example: 'The pen' becomes 'pen' with @def attribute.

(2) (be,,.@verb),(,,.@verb.@progressive):=(0,,,,),(2,,.@present,%1.TENSE,) – This rule combines a 'be' verb and the gerund. For example: 'is playing' becomes 'play' with the tense attribute of the 'be' verb added to it.

(3) (some,,),(,,.@noun):=(0,,,,),(1,qua,%1,%2,) – This is a relationship forming rule. When the first UW has the head word 'some' and the second one has the attribute @noun, a "qua" relationship is formed between them. '1' indicates the type of rule, which is relationship formation.

(4) (at,,),(,,.@noun):=(0,,,,),(1,plc,) – This rule forms a "plc" relationship if the preposition 'at' is present before a noun. '0' type indicates the deletion of the entire UW. So here, 'at' would not be present in the final UW list. There are some catches with this rule where it would result in "tim" relationship as in "at noon". This can be taken care of by writing a more specific rule using Ontology (like using (icl>time)) instead of the generic @noun.

Similar to the above, other rules are added to handle all the 46 relations of the UNL as defined by the UNDL.

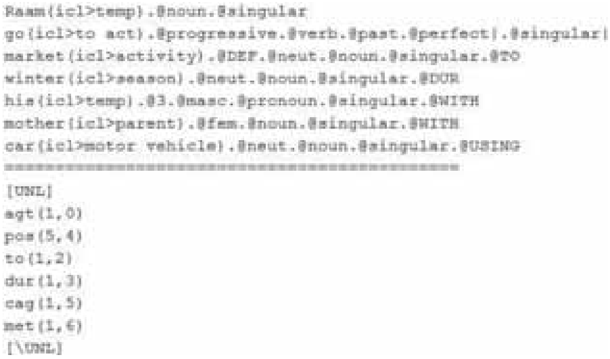
### 4.6 *UNL enconversion algorithm*

As already discussed, the UNL rules are used to convert a given natural language to UNL representation during the enconversion process. The algorithm largely resembles the one followed by UNDL, the difference being that we look at only two nodes at a time, i.e., there is only one analysis and one condition window. The entire set of two rules is split into three groups of rulesets based on their order of execution. Ruleset-1 would be executed first, followed by the other two in order. This facilitates a proper representation, to aid in the deconversion process. Potentially ruleset-1 contains rules which combine adjacent nodes (left composition and right composition). Ruleset-2 contains rules which form relationships between adjacent nodes (left modification, right modification and attribute changing). Ruleset-3 consists of rules which form the compound relationships like and, but, or, etc.

Initially, the first two nodes are taken. Having two nodes, ruleset-1 is first checked to verify whether there is a match with any rule. 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 the case of simple sentences, the process 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. A stage will be reached after using all the three rulesets, where only one node remains in the UW list. This indicates the construction of the final UNL graph. 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 the final list of UWs, and also in the formation of relationships.

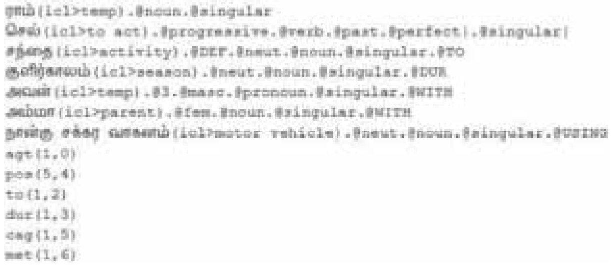
Using the above modified algorithm, the output of the enconversion is given below screen for the example considered.



```
Raam(icl>temp).@noun.@singular
go(icl>to act).@progressive.@verb.@past.@perfect|.@singular|
market(icl>activity).@DEF.@neut.@noun.@singular.@TO
winter(icl>season).@neut.@noun.@singular.@DUR
his(icl>temp).@3.@masc.@pronoun.@singular.@WITH
mother(icl>parent).@fem.@noun.@singular.@WITH
car(icl>motor vehicle).@neut.@noun.@singular.@USING
=============================================
[UNL]
agt(1,0)
pos(5,4)
to(1,2)
dur(1,3)
cag(1,5)
met(1,6)
[\UNL]
```

## 4.7 *Tamil word extractor*

After constructing the UNL graph, the English sentence is now represented in an intermediate form. The translation process should refer to this graph, and convert to the Tamil language and is referred to as deconversion. The deconversion is described in sections 4.7–4.9 to form correct Tamil sentences from the UNL representation. In order to do this, the first step is to retrieve the Tamil word corresponding to every head word (HW) in the UNL. As discussed in the previous section, Tamil words are inserted into the English dictionary manually. Thus, the dictionary is used as a UW dictionary. The query to retrieve the Tamil word matches the HW; and the Tamil word, along with its domain constraint and POS, is retrieved. Only the root word is extracted here. For HWs whose equivalent Tamil word is not found in the dictionary, Phonetic Transliteration has been done in this work, 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 mapping during the process of transliteration. The following is the output along with its transliteration for the example sentence.



```
ராம்(icl>temp).@noun.@singular
செல்(icl>to act).@progressive.@verb.@past.@perfect|.@singular|
சந்தை(icl>activity).@DEF.@neut.@noun.@singular.@TO
குளிர்காலம்(icl>season).@neut.@noun.@singular.@DUR
அவன்(icl>temp).@3.@masc.@pronoun.@singular.@WITH
அம்மா(icl>parent).@fem.@noun.@singular.@WITH
நான்கு சக்கர வாகனம்(icl>motor vehicle).@neut.@noun.@singular.@USING
agt(1,0)
pos(5,4)
to(1,2)
dur(1,3)
cag(1,5)
met(1,6)
```

Transliteration of the above text

1. Ram
2. sel
3. sandhai
4. kulirkaalam
5. avan
6. amma
7. nangu sakkara vaaganam

## 4.8 *Morphological generator*

After determining the HW for every English word in the UNL graph, complete Tamil words have to be formed, by adding suffixes based on the UNL relationship tag. This module forms the complete Tamil words for the root words extracted. The two main components are the formation of verbs and the formation of nouns. The formation of complete verbs requires the addition of the tense marker and the PNG suffix to the root verb. Unlike English, case ending in Tamil is morphologically instantiated. Example, the noun வீடு (veedu), which means "House" may take the form வீட்டில் (viittil), which means "in the House", in some cases where இல் (il) is the case suffix. In English, the noun "House" will not directly get any case suffix. In Tamil, the formation of complete nouns requires the addition of a plural suffix and case marker. Case markers are added to the nouns depending on the UNL relationship, in which they are involved. In addition, the set of rules, which decide the output when two Tamil words combine, are also written (punarchi rules). For example, these rules are required to determine the output word when suffixes are added to the root.

Examples of such rules are:

(1) If the root noun ends in 'm' and the suffix added starts with a uyir ezuthu, 'm' is deleted and 'thth' is added.

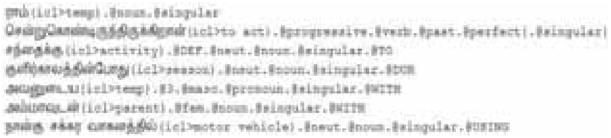
One ‘th’ gets combined with the uyir ezuthu in the suffix and the other ‘th’ remains as such.
For example: ‘puththakam’ + ‘ai’ = ‘puththakaththʼ + ‘ai’ = ‘puththakaththai

புத்தகம் + ஐ = புத்தகத்த் + ஐ = புத்தகத்தை

(2) If the root verb ends in the uyir ezhuthu (vowel), ‘i’, ‘I’, or ‘ai’ and the suffix also starts with a uyir ezhuthu, then a mei (consonant) ‘y’ gets added in between when they combine.
For example: ‘vaanuurthi’ + ‘ai’ = ‘vaanuurthi’ + ‘y’ + ‘ai’ = ‘vaanuurthiyai’

வானூர்தி + ஐ = வானூர்தி + ய் + ஐ = வானூர்தியை

The following is the output for our sample sentence along with its transliteration.



```
ராம்(icl>temp).@noun.@singular
சென்றுகொண்டிருந்திருக்கிறான்(icl>to act).@progressive.@verb.@past.@perfect(.@singular|
சந்தைக்கு(icl>activity).@DEF.@neut.@noun.@singular.@TO
குளிர்காலத்தின்போது(icl>season).@neut.@noun.@singular.@DUR
அவனுடைய(icl>temp).@3.@masc.@pronoun.@singular.@WITH
அம்மவுடன்(icl>parent).@fem.@noun.@singular.@WITH
நான்கு சக்கர வாகனத்தில்(icl>motor vehicle).@neut.@noun.@singular.@USING
```

Transliteration of the above text

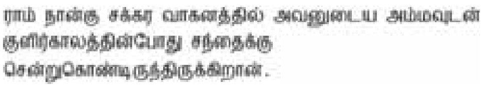
1. Ram
2. sendrukondirundhirukkinraan.
3. sandhaikku
4. kulikaalathinpodhu
5. avanudaya
6. ammavudan
7. nangu sakkara vaaganathil

### 4.9 *Sentence formation*

After adding the suffixes to the noun and the verb forms of the root words, the words need to be framed into a sentence. Tamil grammar is used for the creation of Tamil sentences. Tamil sentences mostly follow the SOV order [29]. 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 is formed just by concatenation, i.e., the subject comes first, the phrase in the reversed order, then the verb, and this might be followed by any number of similar patterns. In the case of compound/complex sentences, Tamil words corresponding to a conjunction, like mattrum(and), aanal(but) (மற்றும், ஆனால்) etc., are inserted.

Thus in our work, we have used the Tamil grammar rules to add suffixes for the root words and have designed a new sentence formation algorithm. The root words from the UNL graph are obtained and using the SOV pattern a simple sentence in Tamil is formed by adding appropriate suffixes to those words.

The following is the output for the sample sentence that has been considered.



ராம் நான்கு சக்கர வாகனத்தில் அவனுடைய அம்மவுடன்
குளிர்காலத்தின்போது சந்தைக்கு
சென்றுகொண்டிருந்திருக்கிறான்.

Ram nangu sakkara vaaganathil (car), avanudaya ammavudan (with his mother) kulikaalathinpodhu (during winter) sandhaikku (to the market) sendrukondirundhirukkinraan (had been going).

## 5. Results and discussion

### 5.1 *Dataset*

It is obvious that MT using UNL is scalable. In order to test the effectiveness of this system, we tested on 500 sentences from CBSE English textbooks. The sentences contained most of words that we have added to the dictionary and some were not. The system translated the words correctly if it is found in the dictionary and transliterated other words which were not in the dictionary. The dictionary had approximately 500 Tamil words inserted into it. The test data consisted of various types of sentences like simple, compound and complex. Reference translations against which the machine translated outputs were scored have been estimated by looking at the translations of a group of 100 users, age between 20 and 22. The results are summarized below. Some examples of the translation output along with its transliteration are given below.

| | |
|---|---|
| (1) Input: | I have a red pen |
| Reference (human): | நான் ஒரு சிவப்பு பேனா வைத்திருக்கிறேன் /<br>நான் ஒரு சிவப்பான பேனா வைத்திருக்கிறேன்<br>Naan oru sivappu penaa veithuirukkiren /<br>Naan oru sivappaana penaa veithuirukkiren. |
| Our system: | நான் ஒரு சிவப்பான பேனாவை வைத்திருகிறேன்.<br>Naan oru sivappaana penaavai veithuirukkiren. |
| Google translate: | நான் ஒரு சிவப்பு பேனா வேண்டும்.<br>Naan oru sivappu penaa vendum. |

In this example, the reference translation is almost close to our system’s output. However, Google translate, translated to Tamil which meant “I am want a red pen”.

| | |
|---|---|
| (2) Input: | Mittu came to the tree and ate the mango. |
| Reference (human): | மிட்டு மரத்துக்கு வந்து மாம்பழம் சாப்பிட்டார் / மிட்டு மரத்துக்கு வந்து மாம்பழம் உண்டார்<br>Mittu marathukku vandhu maambazham saapittaar / Mittu marathukku vandhu maambazham undaar |
| Our system: | மிட்டு மரத்துக்கு வந்தார் மற்றும் மாம்பழத்தை உண்டார்.<br>Mittu marathukku vandhaar mattrum maambazhathai undaar |
| Google translate: | Mittu மரம் வந்து மாம்பழ சாப்பிட்டேன்.<br>Mittu maram vandhu maambazha saappitten. |

In this example, the synonym for the word "ate" has been chosen by our system. One of the reference translations also used this as the translated sentence. For evaluation, if the meaning of the translated word is correct, we considered that as correct translation even if it is a synonym of the word. The word, "Mittu" is a person and we have transliterated in our system, while Google translate left that word as it is without changing. Since, the input English sentence can be considered as a compound sentence due to the word "and", our translation added the conjunction, "மற்றும்". Google translate on the other hand produced a morphologically incorrect sentence, without even translating the correct noun "Mango".

| | |
|---|---|
| (3) Input: | I am going to Mumbai. |
| Reference (human): | நான் மும்பைக்கு சென்றுகொண்டிருக்கிறேன் / நான் மும்பைக்குச் செல்கிறேன்<br>Naan mumbaikku sendru kondirukkiren / Naan mumbaikku selgiren. |
| Our system: | நான் மும்பைக்கு சென்றுகொண்டிருக்கிறேன்.<br>Naan mumbaikku sendru kondirukkiren |
| Google translate: | நான் மும்பை போகிறேன்.<br>Naan mumbai pogiren. |

In this example, the translated word for "going" has been replaced by synonyms "சென்று", "போ" by our system and Google translate respectively.

| | |
|---|---|
| (4) Input: | He had carried many heavy sacks of corn from the farm to the factory. |
| Reference (human): | அவன் பண்ணையிலிருந்து, தொழிற்ச்சாலைக்கு பல சோள மூட்டைகளை எடுத்துசென்றான் / அவன், தொழிற்ச்சாலைக்கு பண்ணையிலிருந்து பல சோளமூட்டைகளை எடுத்துசென்றான்.<br>Avan pannaiyilirundhu thozhirchalaikku pala chola moottaigalai eduthu sendraan / Avan thozhirchalaikku pannaiyilirundhu pala chola moottaigalai eduthu sendraan. |
| Our system: | அவன் தொழிற்சாலைக்கு பண்ணையிலிருந்து சோளத்தை பல மூட்டைகளை கனமான தூக்கிருந்தான்.<br>Avan thozhirchalaikku pannaiyilirundhu cholaththai pala moottaigalai ganamana thookkirundhaan. |
| Google translate: | அவர் தொழிற்சாலைக்கு பண்ணை இருந்து சோளம் பல கனரக சாக்குகளில் மேற்கொள்ளப்படும்<br>Avan thozhirchalaikku pannaiyirundhu cholam pala ganaraga sakkugalil merkollapadum. |

In this example, the free word order nature of Tamil language shows the difference in the translations of the Reference as well as our system. Google translate output is semantically incorrect. For evaluation, the free word order nature is considered and hence if the sentence is syntactically correct, we considered as a positive score.

## 5.2 *Performance evaluation*

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

5.2a *BLEU score*: Bilingual evaluation understudy (BLEU) is an algorithm for evaluating the quality of a machine translated text. It is used to determine, the closeness of the machine translated output to a human translation. The value of the BLEU score lies between 0 and 1. Values closer to 1 indicate better translations, which is an indication that it is closer to human translation.

(a) *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 the edit distance [42]. This is done to take into account the minor spelling mistakes in Tamil (like sandhi) in contrast to English which does not have one. Sandhi is an additional consonsant which is added to give phonetic stress to a word. Example, consider, “கடைக்குச்சென்றான்“ (kadaikku sendraan), where ‘ச் is the sandhi which gives the additional stress to the word ‘கடைக்கு’.

The BLEU score is computed as given below by calculating the Levenshtein distance. Levenshtein distance is a metric used to determine the difference between two given strings [http://en.wikipedia.org/wiki/Levenshtein_distance]. The Levenshtein distance between two words is the minimum number of single-character edits (insertion, deletion, substitution), required to change one word into the other. Mathematically Levenshtein distance can be expressed as

$$\mathrm{lev}_{a,b}(i,j) = \begin{cases} \max(i,j) & \text{if } \min(i,j) = 0, \\ \min\begin{cases} \mathrm{lev}_{a,b}(i-1,j)+1 \\ \mathrm{lev}_{a,b}(i,j-1)+1 \\ \mathrm{lev}_{a,b}(i-1,j-1)+1_{(a_i \neq b_j)} \end{cases} & \text{otherwise} \end{cases}$$

where $a$, $b$ are the words to be compared and $i$, $j$ are the indices.

- Every word of the candidate translation is compared with every word of the reference translation, which has a common prefix of a length of at least three. Synonyms of words if it were found in the sentence, is considered as a correct translation.
- The Levenshtein distance (also referred to as edit distance) is computed for these two strings (words) using the formula mentioned above.
- This Levenshtein distance is summed up and divided by the total number of words in the sentence to obtain the average and this was considered as the BLUE score of the sentence.

(b) *Results of the BLEU computation*: We obtained a BLEU score of 0.581 using this methodology. The BLEU score was calculated for the output of the Google translation system [http://translate.google.com] with the same corpus, and the score obtained was 0.3489. This low score indicates the ineffectiveness of the Example Based approach to MT, which is being used by the Google translation. Thus, the UNL approach to Machine Translation is generic and works better than the existing approaches. This higher BLEU score is attributed to the efficiency of the enconversion rules and the Morphological Generator. We also analysed the BLEU score by grouping the sentences in the dataset into simple, compound and complex sentences. The values are listed in table 1.

It can be seen from table 1 that the BLEU score is high for simple sentences and decreases further for compound and complex sentences. This decrease in the BLEU score is mainly due to the inadequacy of the enconversion rules.

**Table 1.** BLEU scores for different types of sentences.

| | BLEU score | | |
|---|---|---|---|
| Translation system | Simple | Compound | Complex |
| Our machine translation | 0.588 | 0.557 | 0.498 |
| Google translation | 0.262 | 0.457 | 0.452 |

The rules to handle phrases, clauses, certain kinds of compound sentences, word sense disambiguation, etc., have not been added to our UNL KB. The inadequacy of the words in the dictionary and the errors produced by the morphological generator, also contribute a small percentage to the lower BLEU score. The errors in the POS tagger in terms of not performing word sense disambiguation also percolate through the system, and corrupt the output, which in turn, contributed to the reduced BLEU score.

The higher BLEU score for compound and complex sentences in contrast to simple sentences, in the case of the Google translation is due to the example based approach. The presence of commonly occurring phrases and clauses in the corpus causes the translation to be better in an example based MT. But the results clearly show that by adding enconversion rules, the BLEU of our UNL based MT system can be shown to be much higher than the BLEU of the Google translation for compound and complex sentences.

We analysed the BLEU scores that have been achieved by various approaches to machine translation. The literature survey shows that a BLEU score of 0.6950 has been obtained, using the context based machine translation (CBMT) approach for Spanish, Arabic, and Chinese to English [24]. However, the CBMT approach cannot be used for Tamil due to the characteristics of the language. Being a corpus based approach, this method requires huge backend corpora, which adds to its ineffectiveness.

The English to Hindi MT has been used UNL, achieved a BLEU score of 0.26 [39]. They had tested their system on 60 sentences taken from the agricultural corpus. When our MT system is tested with a technical corpus, it is expected that the BLEU score will reduce, because of the inadequacy of technical words in the dictionary, and the inadequacy of the rules to handle more complex sentences. Hence, we conclude that our system is scalable, by adding more enconversion rules, adding Tamil words to the dictionary, and that our UNL based approach can be evaluated to a higher BLEU score, for any type of corpus. An enhancement of the morphological generator would also attribute to the increase of the BLEU score. Other methods of MT report a BLEU score of 0.16–0.20, thus showing the superiority of the UNL approach.

5.2b *Fluency and adequacy*: The UNL based system was also tested for the measure of fluency and adequacy. These are human evaluated scores, which determine the efficiency of Machine Translation.

**Table 2.** Scale to evaluate fluency. *Source*: This scale is devised by us.

| Score | Level | Description |
|---|---|---|
| 5 | Perfect | Good grammar |
| 4 | Fair | Understandable, minor grammatical errors |
| 3 | Acceptable | Understandable, flawed grammar |
| 2 | Bad | Broken, understandable with effort |
| 1 | Poor | Not understandable |
| 0 | Nonsense | Incomplete, makes no sense |

**Table 3.** Scale to evaluate adequacy. *Source*: This scale is devised by us.

| Score | Level | Description |
|---|---|---|
| 5 | All | No loss of meaning |
| 4 | Most | Most of the meaning conveyed |
| 3 | Acceptable | Noun and Tense information conveyed |
| 2 | Some | Noun information alone conveyed with proper gender |
| 1 | Few | Noun information alone conveyed with wrong gender |
| 0 | None | No meaning conveyed |

Fluency refers to the degree to which, the sentences of the target language are well formed, according to the rules of the 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 the original sentence is also communicated in the translated sentence.

To measure the fluency and adequacy, the input and the translated output sentences of our translation system as well as the Google translation were given to around 60 people in the age group of 20–60 years. A scale of 0–5 was used for both these measures. The description of our created fluency scale and the adequacy scale is shown in table 2 and table 3 respectively.

Table 4 gives the average score of fluency and adequacy for our MT system as well as the Google translation. 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 3 and figure 4.

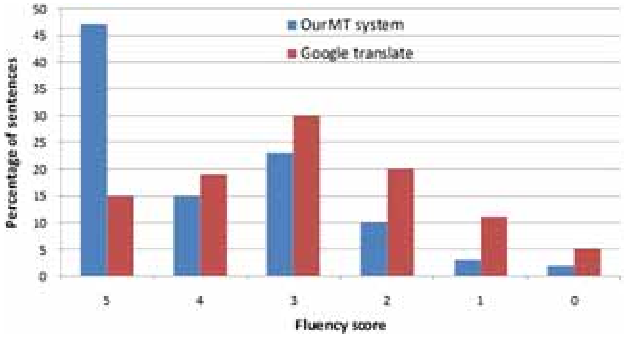


**Figure 3.** Bar chart showing the percentage of sentences under each category of the fluency score.

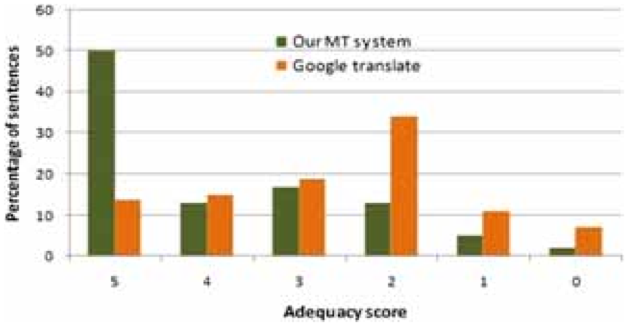


**Figure 4.** Bar chart showing the percentage of sentences under each category of the adequacy score.

We considered the free word order nature of Tamil language for evaluating fluency and adequacy. If the sentence is syntactically correct, the users considered that as a correct translation if all the words, tense and gender were correctly translated. 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 few sentences with a score of 0, 1 and 2 for the adequacy value. On the contrary, the Google translation did not convey tense and gender information for most of the sentences, as seen from figure 3. Moreover, the ordering of words in the Google translation was inefficient, which resulted in the loss of meaning in the target sentence. Figure 4 indicates this loss in meaning, with the adequacy 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 in helping to form grammatically correct sentences.

**Table 4.** Results of the fluency and adequacy tests.

| | Fluency score | | Adequacy score | |
|---|---|---|---|---|
| Translation system | Age (20–40) years | Age (above 40) years | Age (20–40) years | Age (above 40) years |
| Our machine translation | 3.957 | 3.867 | 4.011 | 3.833 |
| Google translation | 2.96 | 2.9 | 2.71 | 2.6 |

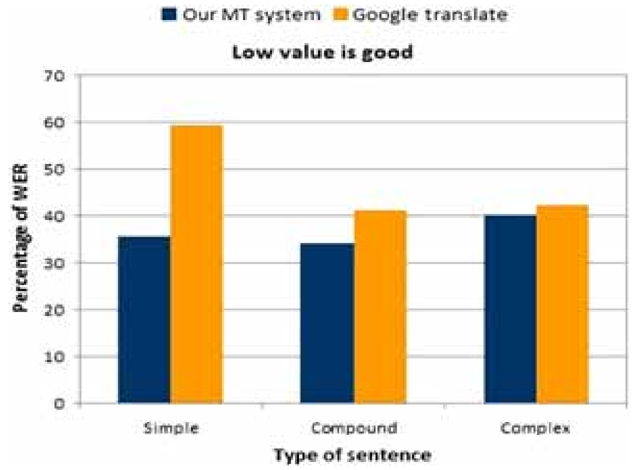


**Figure 5.** Bar chart showing the percentage of WER for different types of sentences.

These results show that the UNL approach helps to retain the meaning of the input sentence, and to achieve grammatical correctness. These high scores are also attributed to the rules of the morphological generator, as they play a major role in the formation of words of the target sentence. Higher scores of fluency and adequacy can be achieved for the UNL system, by improving the enconversion rule base, and increasing the efficiency of the morphological generator.

5.2c *Word error rate (WER)*: WER is a common metric to evaluate the performance of an 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 at least three. As discussed already, the Levenshtein distance essentially gives the number of additions, deletions and modifications of words between the candidate and reference translations.

WER is computed as

$$\textit{Word Error Rate WER} = \frac{S + D + I}{N} \tag{1}$$

where $S$ is the number of substitutions, $D$ the number of deletions, $I$ the number of insertions, and $N$ the number of letters in the reference word.

WER is calculated for every word in the sentences of the candidate translation. If the synonym of the word is present that is not considered as substitution. Insertion results due to the presence of "and" and other conjunctions present in the simple English sentence. The results are then averaged over all the sentences in the corpus. WER for our system using the UNL approach was calculated to be 35.6%, whereas for the Google translation it was 59%. Figure 5 gives the comparison of WER for simple, compound and complex sentences in the dataset, for our translation system as against the Google translation.

The data in figure 5 clearly shows the advantages of incorporating a morphological generator in the system. The Google translation just uses a corpus based approach, and hence, complete sentences are not formed that convey the tense and gender of the subject.

These results show that our newly created morphological generator is quite competent, in that it generates words much closer to the ones in the reference translation. However, the morphological generator is prone to a few sandhi errors and also some conjunction errors. For example, when the suffix இல் ('il') is added to the noun ஆறு ('aaRu'), it gives ஆறில் ('aaRil') instead of ஆற்றில் ('aaRRil'). Errors in the POS tagger also caused wrong suffixes to be added to the root word. The efficiency of our translation system could be improved to reduce the WER, by further enhancing the rules in the morphological generator.

## 6. Conclusion and future work

In this work, we have designed a UNL based machine translation system which translates an English sentence to Tamil. The UNL helps to retain the meaning of the input sentence and also to present the output sentence, in a readable form by utilizing the relationships of the 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. A 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 new sentence formation algorithm produces a meaningful Tamil sentence using the morphologically marked Tamil words.

Certain modifications were made to the standard UNL representation, to help in the process of deconversion. One such modification involves adding information about the gender, person and number in the verb node of the UNL. This is essential since the construction of a complete Tamil verb requires the 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 the performance evaluation are very encouraging and show that the UNL approach to MT resulted in preserving the syntax and semantics of the translation for 82% of the sentences, by achieving an average BLEU of nearly 0.6. The results of the fluency and adequacy tests reveal that the morphological generator is 85% efficient in forming correct words based on the extracted root words and suffixes from the UNL graphs. The new sentence formation algorithm resulted in forming correct sentences with SOV pattern 90% of the time when it was given with suffix added root words.

The errors in the POS tagger propagate down to all the modules of the system, and were instrumental in corrupting the meaning of the translated output. Most of the errors in the system have been found, due to an incorrect POS. A few errors have been found in the spelling of the target words, and adding of the suffixes to root words. These are due to errors in the morphological generator, as this was unable to handle exceptions that are specific to the 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, and punctuation marks (comma, question mark, exclamation mark), and sentences which contain numerals. This shows that more rules should be added to the enconversion process.

The efficiency of the Stanford POS tagger is only around 56% for sentences. Hence, it could be replaced by a better POS tagger, which would handle word sense disambiguation, thus resulting in an improved translation system. Alternatively an additional word sense disambiguation module can be implemented which will reduce POS errors. Parsing of the 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. The ordering of rules in the morphological generator could be looked upon for handling Tamil grammar exceptions. A technique to handle homonyms and numerals needs to be identified. Thus, many improvements and extensions are possible, and this system offers a wide scope in the field of machine translation.

## References

[1] James F Allen 2003 *Natural language processing*

[2] William John Hutchins and Harold L Somers 1992 *An introduction to machine translation*

[3] Jonathan Slocum 1985 A survey of machine translation: its history, current status, and future prospects. *Comput. Linguist.* 11(1): 1–17

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

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

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

[7] Bhattacharyya P, Hegde J, Shah R M, Ramanathan A and Sasikumar M 2008 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

[8] Richard Zens and Hermann Ney 2004 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

[9] Hua Wu and Haifeng Wang 2007 Pivot language approach for phrase-based statistical machine translation. *Mach. Transl.* 21(3): 165–181

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

[11] Constantine Domashnev, Nirenburg Sergei and Dean J Grannes 1993 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

[12] Bar Kfir, Choueka Y and Dershowitz N 2007 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

[13] Harold Somers 1999 Review article: Example-based machine translation. *Mach. Transl.* 14(2): 113–157

[14] Konstantin Tretyakov 2007 Example-based machine translation of short phrases using the context equivalence principle

[15] Daniel Jones 1992 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

[16] Yves Lepage and Etienne Denoual 2005 Purest ever example-based machine translation: Detailed presentation and assessment. *Mach. Transl.* 19(3): 251–282

[17] Michael Carl, Andy Way and Walter Daelemans 2004. Recent advances in example-based machine translation. *Comput. Linguist.* 30(4): 516–520

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

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

[20] Ruvan Weerasinghe 2003 A statistical machine translation approach to Sinhala-Tamil language translation. *Towards an ICT enabled Society*, pp. 136–141

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

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

[23] Margaret King, Hovy Eduard and Andrei Popescu-Belis 2002 Principles of context-based machine translation evaluation. *Mach. Transl.* 17(1): 43–75

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

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

[26] Tynovsky M 2008 Hybrid approaches in machine translation. In: *WDS Proceedings of Contributed Papers, Part-I.* pp. 124–128

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

[28] Michael Carl, Cathrine Pease, Leonid L Iomdin and Oliver Streiter 2000 Towards a dynamic linkage of example-based and rule-based machine translation. *Mach. Transl.* 15(3): 223–257

[29] Thenmozhi D and Aravindan C 2009 Tamil-English cross lingual information retrieval system for agriculture society. In: *Proceedings of the International Forum for Information Technology in Tamil*. pp. 173–178

[30] Saraswathi S, Anusiya M, Kanivadhana P and Sathiya S 2011 Bilingual translation system for weather report. In: *Proceedings of the International Conference on Advances in Computing and Communications*. pp. 155–164

[31] Sohail Asghar Tahir, Ghulam Rasool and Nayyer Masood 2010 Knowledge based machine translation. In: *Proceedings of the IEEE International Conference on Information and Emerging Technologies*. pp 1–5

[32] Sergei Nirenburg 1989 Knowledge-based machine translation. *Mach. Transl.* 4(1): 5–24

[33] Tarcisio Della Senta, Hiroshi Uchida and Meiying Zhu 2005 *Universal networking language*. UNDL Foundation, Tokyo, Japan

[34] Amitabha Mukerjee, Achla M Raina, Kumar Kapil, Pankaj Goyal and Pushpraj Shukla 2003 Universal networking language: A tool for language independent semantics? *Univ. Netw. Lang.: Adv. Theory Appl.* (2003) 145–150

[35] Jignashu Parikh, Dave Shachi and Pushpak Bhattacharyya 2001 Interlingua-based English Hindi machine translation and language divergence. *Mach. Transl.* 251–304

[36] Hameed M S, Subalalitha C N, Geetha T V and Parthasarathi R 2012 A deconverter framework for Malayalam. In: *Proceedings of the International Conference on Advances in Computing, Communications and Informatics*. ACM, pp. 847–856

[37] Kumar P and Sharma R K 2013 Punjabi DeConverter for generating Punjabi from Universal Networking Language. *J. Zhejiang Univ. Sci. C* 14(3): 179–196

[38] Arif MdA 2011 Problems and prospects: Universal Networking Language on Bangla sentence structure perspective. *Int. J. Eng. Tech.* 11(4): 147

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

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

[41] Balaji J, Geetha T V, Parthasarathi R and Karky M 2011 Morpho-semantic features for rule-based Tamil enconversion. *Int. J. Comput. Appl.* 26: 11–18

[42] Michael Gilleland 2005 Levenshtein distance. Three flavors. http://www.merriampark.com/ld.htm