A Hybrid Approach to English to Malayalam Machine Translation

Nithya B
PG Student
Dept. of CSE
Government Engineering College,
Thrissur, India

Shibily Joseph
Assistant Professor
Dept. of CSE
Government Engineering College,
Thrissur, India

ABSTRACT
Machine translation is the process of translating text from one natural language to other using computers. The process requires extreme intelligence and experience like a human being that a machine usually lacks. Availability of machine translators for translation from English to Dravidian language, Malayalam is on the low. A few corpus-based and non-corpus based approaches have been tried in performing English to Malayalam translation. In this work a hybrid approach to perform English to Malayalam translation is proposed. This hybrid approach extends the baseline statistical machine translator with a translation memory. A statistical machine translator performs translation by applying machine learning techniques on the corpus. The translation memory caches the recently performed translations in memory and eliminates the need for performing redundant translations. The system is implemented and evaluated using BLEU score and precision measure and the hybrid approach is found to improve the performance of the translator.

General Terms
Translation, Machine Learning

Keywords
Hybrid Machine Translation, Statistical Machine Translation, Translation Memory, Corpus, English-Malayalam Translation

1. INTRODUCTION
Machine Translation (MT) is the use of computers to automate some or all of the process of translating from one language to another [1]. It is one of the widely researched tasks and is a subfield of Natural Language Processing. Manual translations are always time consuming and expensive. The use of machine translators enables quick and easy creation of content with a reduced manual effort. Even though MT technology promises fast translations, they are not easy to implement.

A machine translator usually deals with two languages namely a source language and a target language. Source language refers to the language that is to be translated and target language refers to the language to which the source language is translated. Thus the input to a translator is some source language text and the translator generates the equivalent target language text as the output.

Machine translators can be bilingual or multilingual. A bilingual machine translator converts from one language to another whereas multilingual translators can translate languages from a specified set of languages to another set of languages.

MT has great significance in India due to the large number of regional languages in the country. At least 30 different languages and around 2000 dialects have been identified. Majority of the population in the country is not good in handling English. Malayalam is a Dravidian language with about 38 million speakers spoken mainly in the south west of India, particularly in Kerala. It is one of the 22 scheduled languages of India and was declared a classical language by the Government of India in 2013. The electronic resources for the language are quite low. Also the availability of machine translators for English to Malayalam language pair is low.

There are different techniques for machine translation. The major machine translation techniques are Rule Based Machine Translation (RBMT), Statistical Machine Translation (SMT) and Example based machine translation (EBMT). Development of rule based systems is expensive and time-consuming. They also require extensive knowledge about the language features. Now SMT is the dominant approach in the field. SMT approach is being used by Google, Microsoft etc. in their online translation systems. SMT systems require minimal linguistic knowledge and minimal human effort. So it is easy to build an SMT system for a new pair of language with reduced cost and time. The only prerequisite for this is availability of training data.

A Translation Memory(TM) is a system which can reuse the previous translations. For a previously translated sentence, the TM system provides the previously translated output. The TM analyzes the source text and only sends those sentences to the machine translation system for which there is no translation available. A TM thus eliminates the need for performing redundant translations.

In this work an English to Malayalam machine translator is built using a hybrid approach. This hybrid technique combines TM with a statistical machine translator.

The rest of this paper is organized as follows. Section 2 covers the related works. Section 3 gives an overview about the proposed system. Section 4 gives the implementation details. Section 5 gives the evaluation of the new system and section 6 concludes the paper.

2. RELATED WORKS
Machine translation research activities started in India in mid 80s and early 90s. Only a few efforts are there towards the building of English to Malayalam machine translators. A rule based translation approach for English to Malayalam was proposed in [2]. In this method translation was mediated by English to Malayalam bilingual dictionaries and rules for converting source language structures into target language

structures. The rules used in this approach were prepared based on the Parts Of Speech (POS) tag and dependency information obtained from the parser. The system could translate English sentences to their Malayalam equivalent. Limitation was that only sentences with length upto six could be handled by this system.

A methodology for translating English sentences to equivalent Malayalam using statistical model was proposed in [3]. The system used a monolingual Malayalam corpus and a bilingual English-Malayalam corpus in the training phase. By applying machine learning techniques in the corpus, translations could be generated. A technique to improve the alignment model by incorporating the parts of speech information into the bilingual corpus was also proposed. Suffix separation was applied on the Malayalam corpus and stop word elimination was applied on the bilingual corpus to make the translation better. The system could generate translations of fairly good quality.

Development of an SMT system for English to South Dravidian languages like Malayalam and Kannada by incorporating syntactic and morphological information was proposed in [4]. The system was able to perform good translation even for simple sentences having more than ten words.

AnglaMalayalam machine translator was developed for English to Malayalam translation by a consortium of Indian Universities along with CDAC and IIIT jointly. The project was funded by TDIL (Technology Development for Indian Languages) and MHRD. The project was part of the multilingual AnglaMT system and used an interlingua based method.

Integration of Translation Memory(TM) with AnglaMT, a RBMT, was proposed in [5]. This work was based on the fact that an MT system with human intervention was the most appropriate way to get translation which has greater human acceptability. Human machine unification was tried. Wordfast a Computer-Aided Translation (CAT) program designed as a Microsoft Word add-on was integrated with AnglaMT system for English to Hindi and English to Punjabi translation.

3. OVERVIEW OF THE PROPOSED SYSTEM

In this work English to Malayalam translator model is developed using a hybrid approach. The approach is hybrid in the sense that it combines a statistical machine translator with TM. The proposed system provides an extension to the baseline SMT system by introducing a TM. Baseline SMT system is a simple phrase-based statistical machine translator.

The architecture is shown in the Fig. 1. The proposed system has two main components. They are statistical machine translator and translation memory.

3.1 Statistical Machine Translator

A statistical machine translator is built using statistical machine translation approach. Statistical processing of natural language is based on corpus. SMT [6] is a corpus-based machine translation approach in which machine learning techniques are applied to a bilingual corpus to produce a translation system automatically. For an SMT system, a parallel corpus consisting of source and target language sentences and a monolingual corpus consisting of target language sentences are required. The SMT system is trained on these large quantities of parallel data and monolingual data. The statistical model learns the translation parameters from the corpus and performs the translation.

Parallel data is a collection of sentences in two different languages, which is sentence-aligned i.e., each sentence in one language is matched with its corresponding translated sentence in the other language. It is also known as a bitext. From the parallel data the system learns how to translate small segments and from the monolingual data the systems learn what the target language should look like.

The SMT approach can be simply explained as follows. Every sentence in the target language is considered as the translation of a source language sentence with some probability and the best translation is the sentence that has the highest probability

SMT models view the machine translation as a noisy channel model. If we want to translate a sentence *f* in the source language F to a sentence *e* in the target language E, the noisy channel model describes the situation in the following way. Suppose that the sentence *f* to be translated was initially conceived in language E as some sentence *e*. During communication *e* was corrupted by the channel and became *f*. Now assume that each sentence in E is a translation of *f* with some probability and the sentence we choose as the translation is the one that has the highest probability, i.e., the sentence *e* that maximizes the probability $Pr(e|f)$ [7].

The SMT can be split into two main phases:

(1) Training phase

(2) Translation phase

During training phase, a statistical model of translation is built from the corpus. The training phase is itself split into three parts:

(1) Document collection which is the collection of texts which form the corpus

(2) Building the language model for the target language from the monolingual corpus i.e., $Pr(e)$

(3) Building the translation model from the target language to the source language i.e., $Pr(f|e)$

The second phase is the translation phase or the decoding phase, which uses a heuristic search procedure to find a good translation of the given source language text.

An SMT system requires a method for computing language model probabilities, a method for computing translation probabilities, and a method for searching among possible target sentence *e* that gives the greatest value for $Pr(e)Pr(f|e)$.

3.1.1 Language Model

Language model is a statistical model built using monolingual data in the target language and is used to ensure the fluency of the output. For a sentence *e* in the target language, language modelling involves computing $Pr(e)$. For a good sentence in the target language $Pr(e)$ will be high and will be low for bad sentences. Usually N-gram model is used to compute this probability.

3.1.2 Translation Model

Language model is a statistical model built using monolingual data in the target language and is used to ensure the fluency of the output. Given a pair of strings *(f,e)*, translation model

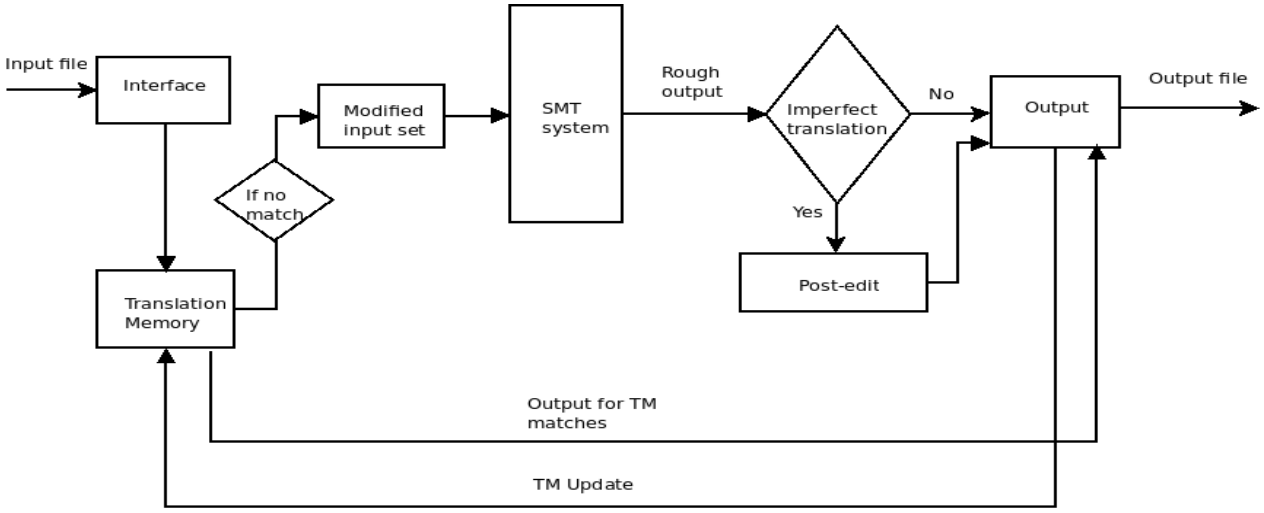

Fig 1: System architecture

computes the probability *Pr(f|e)* by learning from the parallel corpus. This probability *Pr(f|e)* will be high if *f* and *e* are equivalent sentences in the source and target languages respectively.

3.1.3 Decoding

Given a language model, translation model and a sentence *f* in the source language, decoder finds the translation *e* which maximizes the product *Pr(e)Pr(f|e)*. This is a crucial part in SMT. A reliable and efficient decoding algorithm is required to find quickly the highest probability translation among the exponential number of choices.

3.2 Translation Memory

Most of the translation needs are highly repetitive. If it is possible to find existing translations of the source language text, then carrying out redundant translations can be avoided. This idea is exploited in a TM.

A TM is a tool which has the ability to cache recent translations in memory. Thus the system can translate previously seen inputs with the help of previously generated translations which has already been cached. This minimizes the effort and time required for translation. Also this system allows for human intervention in the form of post-editing so that a human expert can make modifications in the translator output. As machine translators may not always produce perfect translations, the human intervention enables to correct mistakes if any.

TM analyzes the input sentence and checks if it is already available in its database. If it finds a match it will use the corresponding translated output. An unseen source sentence is handled by the machine translator itself. Thus TM eliminates the effort for running the translator for a previously translated sentence.

A TM should have four main components. They are

(1) A mechanism to store sentences and their translations

(2) A search mechanism to find input sentence matches from TM

(3) Provision for post-editing the translator output

(4) Provision for updating the TM

3.3 System Flow

The input sentence to be translated is first searched in the TM. If the sentence has a match in the TM, then translation is performed by directly copying the TM match's translation. But if there is no match available in TM, the input is sent to the SMT system. The SMT system performs translation and provides a rough output. This output can be modified by a human expert in case it is not perfect. The final output is fed to the TM too so as to cache the translation.

4. IMPLEMENTATION

4.1 Data Set

The primary requirement for building an SMT system is corpus. This includes English-Malayalam parallel corpus and a monolingual Malayalam corpus. The availability of parallel corpus is quite low for Malayalam language. Hence a small parallel corpus consisting of 563 sentences is used as the training data set. The domain selected is Indian history and Islamic history. A tuning data set consisting of 150 sentences is used for tuning.

4.2 Tools Used

To build the proposed English to Malayalam translator various natural processing tools are available. The SMT system requires three components namely language model, translation model and decoder. The open source tools such as IRSTLM, GIZA++, Moses decoder etc. were used in this work for implementing the proposed system. The tools were installed in Ubuntu 10.04 operating system environment.

4.3 Moses

Moses is a statistical machine translation system that allows to automatically train translation models for any language pair. Moses is an implementation of the statistical approach to machine translation. The two main components in Moses are the training pipeline and the decoder. The training pipeline is a collection of tools (mainly written in Perl, with some in C ++) which take the raw data (parallel and monolingual) and turn it into a machine translation model. The decoder will translate the source sentence into the target language [6].

4.4 GIZA++

GIZA++ is an extension of the program GIZA which was developed by the Statistical Machine Translation team. GIZA++ implements IBM-4 alignment model with a dependency of word classes and also implement an alignment model based on Hidden Markov model. On giving a bilingual parallel corpus to GIZA++, it computes the translation probabilities using an unsupervised learning algorithm called Expectation-Maximization algorithm [8].

4.5 IRSTLM

IRSTLM is an open source language modelling toolkit. IRSTLM toolkit handles LM formats which permit to reduce both storage and decoding memory requirements and to save time in LM loading. LM estimation starts with the collection of n-grams and their frequency counters. Then, smoothing parameters are estimated for each n-gram level, infrequent n-grams are possibly pruned and, finally, a LM file is created containing n-grams with probabilities and back-off weights. This procedure can be very demanding in terms of memory and time if applied to huge corpora. IRSTLM provides a simple way to split LM training into smaller and independent steps, which can be distributed among independent processes With IRSTLM, the binary format of the language model can be generated. This allows LMs to be efficiently stored and loaded. IRSTLM can be integrated into the popular open source SMT decoder Moses [9].

4.6 Transliteration

The Moses tool can handle only languages written in Latin script. But Malayalam language is written in non-Latin script and hence cannot be handled by the Moses directly. Malayalam words written in English are used by Moses in developing the machine translator model for the language. Hence appropriate conversions are accomplished using transliteration tools.

Transliteration maps characters in the source language to characters that have similar pronunciation in the target language. Transliteration preserves word pronunciation. Tools employed for transliteration in this work are SILPA and Google Transliterate.

4.6.1 SILPA Transliteration

Malayalam to English conversion is accomplished by means of SILPA Transliterator for Malayalam to English. SILPA (Swathanthra Indian Language Computing Project) is web framework which has got a set of applications for processing Indian Languages in many ways. The transliterator application helps to transliterate text from any Indian language to any other Indian language.

4.6.2 Google Transliterate

Google Transliterate API is used for performing transliteration from English to Malayalam. This service is accessible via JavaScript API.

4.7 Building a baseline SMT system

The steps involved in building a baseline SMT system are:

4.7.1 Corpus Preparation

Training data is provided in a sentence aligned (one sentence per line) format, in two files, one for the English sentences and one for the Malayalam sentences. English sentences are stored in a file with .en extension and the corresponding Malayalam translations are stored in a file with .ml extension with one sentence per line. The corpus is initially subjected to certain pre-processing steps like tokenization, truecasing and cleaning [6]. As Malayalam is written in non-Latin script, the Malayalam corpus is romanized first before pre-processing.

4.7.2 Language Model Training

Language model training is done on the target language corpus, i.e., Malayalam corpus. The romanized Malayalam corpus is considered. The corpus is preprocessed initially by performing tokenizing, truecasing etc. A 3-gram language model of the Malayalam language is generated using scripts from the IRSTLM distribution.

4.7.3 Training the Translation System

This is the main step in which the training of the translation model takes place. Moses distribution has scripts for training. This training involves performing word alignment using GIZA++, phrase extraction and scoring, creating lexicalised reordering tables and finally generating the configuration file moses.ini. This moses.ini file contains all the correct paths for the generated model and a number of default parameter settings.

4.7.4 Tuning

Tuning is done using a small amount of parallel data, separate from the training data. The tuning data is also subjected to tokenizing and truecasing. Tuning is done using mertmoses.pl script. The tuning script requires the tuning corpus, location of Moses decoder and Moses configuration file. On running this script a new configuration file moses.ini is generated with trained weights.

4.7.5 Testing

This is the final step in which the Moses SMT decoder is run. The Moses decoder performs a heuristic search procedure to find out the translation of the source language sentence. The decoder is run on the test set by invoking the Moses executable and with the configuration file moses.ini as an argument. The output from the SMT system is fed to Google Transliterate service via JavaScript API and the final Malayalam output is generated.

4.8 TM Implementation

TM is implemented in C programming language. The program looks for matches in the input with the TM contents. The exact match case is implemented in this work. An exact match is a perfect character by character match between current source segment and stored source segment. Those sentences in the input file with matches in the TM are translated by looking at the translations in the TM. Only the remaining sentences are fed to the Moses decoder. The initial run of Moses baseline SMT creates the TM. The subsequent runs of the hybrid system make use of this TM. The TM match program finds the sentences already available in TM that matches with the given input set and uses the corresponding translation from the TM. The remaining sentences are fed to the Moses and output is generated. This output is made to undergo post-editing process and the final translations are updated in the TM.

5. EVALUATION

The system was tested with a test set consisting of 70 English sentences. Both manual and automatic evaluation techniques were employed to measure the efficiency of the new approach.

5.1 BLEU Evaluation

The Bilingual Evaluation Understudy (BLEU) evaluation is an inexpensive automatic evaluation that is quick and language independent. It compares n-grams of the candidate with the n-grams of the reference translation and counts the number of matches. The more the matches, the better the candidate translation is. BLEU measures accuracy. Thus larger BLEU scores are better.

The evaluation was done by running multi-bleu.perl script with the reference files and MT output. The output parameters include

(1) BLEU score

(2) Brevity Penalty (BP) = min (1, number of output words/ number of ref words)

(3) ratio = number of output words / number of ref words

(4) hyp_len = number of output words

(5) ref_len= number of ref words.

Here ref denotes reference translations. The evaluation results are tabulated below.

Table 1. BLEU Evaluation

Parameter	Baseline	Hybrid
BLEU Score	68.14	69.33
BP	1.000	1.000
Ratio	1.023	1.021
hyp_len	404	429
ref_len	395	420

The evaluation results shows improvement in BLEU score for the hybrid system compared to the baseline system.

5.2 Manual Evaluation

Manual evaluation is done by taking the opinion of a human expert regarding the quality of the translator output. As part of the manual evaluation, precision of the MT output was calculated. Precision is the ratio of number of sentences for which perfect translation was obtained to the number of input sentences. It will be a value between 0 and 1. The precision for the hybrid system was obtained as 0.753 which indicates that the system has an accuracy of 75.3%.

5.3 Performance Evaluation

The performance evaluation was done based on the time taken for translation. The test set containing 70 sentences got translated into the intermediate form (Malayalam words written in English script) in approximately in 0.513 seconds. The time for transliteration was dependent on the availability and strength of the internet connection which is variable.

6. CONCLUSION

A framework to build a hybrid machine translation system from English to Malayalam using statistical model is proposed and developed. The system provides an extension to a phrase-based SMT system by integrating with TM. This integration speeds up the translation process as the recent translations are cached. The post-editing feature, which allows human intervention, helps to make the translation more perfect. The work can be further extended by adding more sentences to the parallel corpus. Also the system can be implemented in parallel on a large cluster of machines which will be quite effective in processing large quantities of training data.

7. ACKNOWLEDGMENTS

The authors wish to thank the anonymous reviewers for their useful suggestions that helped in improving the quality of this paper.

8. REFERENCES

[1] Dan Jurafsky, James H Martin, Andrew Kehler, Keith Vander Linden, and Nigel Ward. Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition, volume 2. MIT Press, 2000.

[2] Remya Rajan, Remya Sivan, Remya Ravindran, and K.P Soman. Rule based machine translation from english to malayalam. In Conference Proceedings on International Conference on Advances in Computing, Control, and Telecommunication Technologies, pages 439–441, 2009.

[3] Mary Priya Sebastian, G Santhosh Kumar, et al. English to malayalam translation: a statistical approach. In Proceedings of the 1st Amrita ACM-W Celebration on Women in Computing in India, page 64. ACM, 2010.

[4] P Unnikrishnan, PJ Antony, and KP Soman. A novel approach for english to south dravidian language statistical machine translation system. International Journal on Computer Science and Engineering (IJCSE), 2(08):2749–2759, 2010.

[5] Nishtha Jaiswal, Renu Balyan, and Anuradha Sharma. A step towards human-machine unification using translation memory and machine translation system. In International Conference on Languages, Literature and Linguistics, pages 64–68.2011.

[6] Philipp Koehn and Hieu Hoang. Moses. Statistical Machine Translation System, User Manual and Code Guide, 2010.

[7] Ananthakrishnan Ramanathan. Statistical machine translation. Ph. D Seminar Report, 2008.

[8] F. J. Och and H. Ney. Improved statistical alignment models. pages 440–447, Hongkong, China, October 2000.

[9] Marcello Federico, Nicola Bertoldi, and Mauro Cettolo, Irstlm: an open source toolkit for handling large scale language models. In Interspeech, pages 1618–1621, 2008.