Language Localisation of Tamil using Statistical Machine Translation

Y. Achchuthan
Department of Computer Science,University of Jaffna, Sri Lanka.
achch1990@gmail.com

1. ABSTRACT

Language localisation, where the strings in interface and documentation are translated to a new language, is a rigorous and time consuming task. On the other hand machine translation systems, specifically Statistical Machine Translation (SMT) systems, are successfully used among many language pairs. A few SMT systems have been developed for generic domain; however, there are no systems available to aid localisation yet. This research proposes a new methodology in which language localisation can be done using SMT. This research also identifies suitable parameters on which a SMT aided localisation system could be built. A pilot system is developed and the system is also outlined in this paper. A RESTful API has also been developed to facilitate localisation in remote tools. Several open source software have been translated already to Tamil. Those translated English – Tamil pairs were collected from various language resource files and then cleaned, tokenised and were used to train the system. Another similar system is prepared with data from generic domain apart from the collected technical data. Systems were trained with 2-gram, 3-gram and 4-gram language models that are created using two different language modelling tools namely KenLM and IRSTLM. Then the results were evaluated using BLEU algorithm. Appropriate parameters for setting up SMT system for localisation were identified from the evaluation. The results show that it would be enough to train a system with 3-gram, and the modified BLEU algorithm will give better understanding of the results compare to the original implementation of it. Further KenLM was found to perform better than IRSTM in terms of accuracy of results and the speed of execution.

2. INTRODUCTION

Localisation of software has become an inevitable part of software development. Language localisation is part of the software localisation and it is a time consuming and rigorous activity. Software vendors, especially the Open Source software vendors find difficulties in gathering volunteer language localisers. On the other hand the people who have willingness, are also not consistent. Further, different people may come up with different translations for the same set of terms if they use different glossaries. Nowadays, most of the rigorous and time consuming tasks are being automated by using machine. This research explores whether the language localisation can be automated using a concept called SMTThe field of Machine Translation(MT) has a long term history. Several researches have been done to make a MT systems for different languages using different methods such as Rule-based Machine Translation and SMT have been employed so far.

3. METHODOLOGY

This section elaborates the proposed system and the steps involve in the proposed statistical machine translation system. There are several tweaks have been done to the usual SMT steps, importantly in language modelling and in BLEU score evaluation.

1. Proposed System

A web based system (shown in Figure 1) and a SMT system are developed to do language localisation using SMT. The system accepts English language resource file in PO format and output the corresponding Tamil language resource file in PO format.

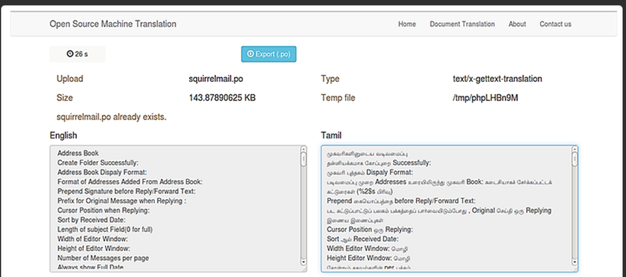

Figure 1: Proposed System

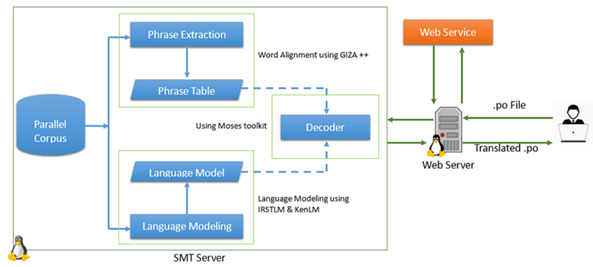

Figure 2: Architecture Overview

A high level architecture of the proposed system is shown in Figure 2. As shown in the system, a web service interface also have been developed through which translation service can be obtained using RESTful way. If a phrase is passed then the system will process it and will return its localisation.

2. SMT System

Though there is few popular SMT development framework, Moses is used to develop the proposed SMT system. Because, there are no primary NLP tools available for Tamil language with reportable accuracy. Corpus preparation, Language modelling, Training, Testing and Evaluation are the key steps of SMT. There are several tools available to carry out these tasks. However, in this research GIZA++, KenLM, IRSTLM are used with the Moses framework to carryout above tasks. Each of these steps are elaborated in details in the following sections.

Figure 2: Flow diagram of the proposed method.

2.1. Corpus Preparation

TABLE 1 : COLLECTED DATA FROM THE INTERNET

Source	Sentences (No. of phrases)
Mozilla Firefox	4568
Mozilla OS	3465
Joomla	4358
Drupal	4544
Moodle	4355
Squirrel Mail	1116
Tamil Glossary	2567
EnTam v2.0	169871

Large amount of parallel text is required to build a SMT system. Further these text should be prepared in such a way that those can be fed to the system. Text are collected and prepared in appropriate format in this phase. This preparation phase consists of sub phases such as data collection, tokenisation, truecasing and cleaning.

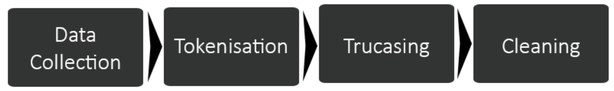

Figure 2: Flow diagram of the proposed method.

2.2 Language Modeling

There are several tools available to do the language modelling. However, most of them use the same algorithm called Kneser-Ney to do the modelling. Some of them support upto 5-gram and others support upto 3-gram language modelling. In this research work, two different language modelling tools have been used to see which one suite better. Two language modelling tools that work with Moses IRSTLM and KenLM are used. Among these, KenLM provides output upto 5-gram and this also support for modelling billions of tokens using big data tools. Since the output language is Tamil, language modelling is only need to be done for Tamil text corpus to find the most probable output. The language modelling were done using both tools for 2-gram, 3-gram and 4-gram. Results were obtained using all these attempts and
documented.

2.3 Training

This is the prime activity of the SMT system, where phrases from English – Tamil parallel corpus are aligned from English to Tamil. There are several algorithms available to do the alignment in different level such as word level and phrase level. This research phrase based alignment is used as it is most appropriate for English – Tamil translation. A tool called GIZA++ is used for this phrase-alignment, which supports for phrase-based alignment and the results can be plugged into Moses framework.

2.4 Testing and Evaluation

The trained SMT system is tested with five different unseen data set or reference translation. It was noted that five reference translations may show better results than using a single reference translation. These tests were performed using two languagemodelling tools and BLEU scores were obtained using the customised equation is shown in Equation 1.

Equation 1: Modified formula for BLEU

$$BLEU = \min\left(1, \frac{output_{length}}{reference_{length}}\right)\left(\prod_{i=1}^{3} precision_i\right)^{\frac{1}{3}}$$

4. RESULTS AND DISCUSSION

It was decided to obtain evaluation results for the systems that trained with 2-gram, 3-gram and 4-gram language models to see which model gives better results. Table 2 shows that the results of the system that is trained using generic corpus and Table 3 shows that the results of the system that is trained using only technical corpus obtained from different open source software.
Also, the results obtained for 5 different data sets is shown in Table 2 and Table 3. 100 unseen phrases were obtained from Firefox, Joomla and Drupal and tested. Further two other data sets were compiled by gathering 400 and 1500 random unseen phrases from the technical data set. Next, the results were obtained for two different language modelling tools namely KenLM and IRSTLM. The results based on the KenLM and IRSTML are shown the following tables.

TABLE 2: BLEU SCORES WITH ACTUAL ALGORITHM AND USING THE GENERIC CORPUS AS THE TRAINING DATA SET

	FF	JMLA	DPL	T400	T1500
2G(KENLM)	14.41	24.80	**56.03**	**12.88**	2.74
2G(IRSTLM)	**14.43**	**25.87**	55.09	12.74	**3.04**
3G(KENLM)	13.39	27.57	**64.09**	**14.36**	**3.38**
3G(IRSTLM)	**13.67**	**27.75**	59.44	13.42	2.91
4G(KENLM)	13.49	27.88	**63.25**	**14.17**	**3.24**
4G(IRSTLM)	**13.57**	**28.28**	59.73	13.64	2.93

FF:-Firefox, JMLA:-Joomla, DPL:-Drupal, T400-Test data with 400 phrases, T1500-Test data with 1500 phrases

TABLE 3: BLEU SCORES WITH ACTUAL ALGORITHM AND USING THE TECHNICAL CORPUS AS THE TRAINING DATA SET

	FF	JMLA	DPL	T400	T1500
2G(KENLM)	**17.97**	25.28	58.48	12.88	**2.78**
2G(IRSTLM)	14.26	**27.79**	**63.20**	**13.38**	2.48
3G(KENLM)	**17.62**	**24.41**	**66.66**	**14.32**	**2.86**
3G(IRSTLM)	14.60	12.38	21.37	6.22	2.15
4G(KENLM)	**17.79**	**26.18**	**66.66**	**14.31**	**2.87**
4G(IRSTLM)	14.60	12.38	21.37	6.22	2.15

FF:-Firefox, JMLA:-Joomla, DPL:-Drupal, T400-Test data with 400 phrases, T1500-Test data with 1500 phrases

5. CONCLUSION

The results show that the software localisation can be done with the aid of Statistical Machine Translation system. To get better performance, a Statistical Machine Translation should be trained with 3-gram language model. KenLM is more suitable compare to the IRSTLM, because, KenLM shows slightly betterresults and gives results more quickly compared to IRSTLM. When the SMT systems for the localisation are compared, customised BLEU implementation gives reasonable results compared to actual BLEU implementation. During the customisation the BLEU algorithm should be tuned to evaluate for 3-gram, instead of 4-gram. This approach may be useful for the language similar to Tamil, for those language localisation also the aided systems can be implemented with these given parameters. The proposed system allows users to upload a resource file in the form of Portable Object and provides the translated text as another Portable Object. The system also has a RESTful API and the phrases can be translated by calling this API, .This API will be useful for other software vendors to implement their software localisation aided system.

6. REFERENCES

[1]. P. Koehn, et al. "Moses: Open Source Toolkit for Statistical Machine Translation," in *the Proceedings of the ACL 2007 Demo and Poster Sessions*, 2007, pp. 177-180.

[2]. L. Ramasamy, O. Bojar, and Z. Žabokrtský, "Morphological Processing for English-Tamil Statistical Machine Translation," in *Proceedings of the Workshop on Machine Translation and Parsing in Indian Languages (MTPIL-2012, COLING)*, Mumbai, India, December 2012, p. 113.

[3]. R. Weerasinghe, "A Statistical Machine Translation Approach to Sinhala-Tamil Language Translation," in *Towards an ICT enabled Society*, 2003, p. 136.

[4]. K. Papineni, S. Roukos, T. Ward, and W. J. Zhu, "BLEU: a method for automatic evaluation of machine translation," in *Proceedings of the 40th annual meeting on association for computational linguistics*, 2002, pp. 311-318.

[5]. P. Koehn, A. Birch, and R. Steinberger, "462 Machine Translation Systems for Europe," in *Proceedings of MT Summit XII*, 2009, pp. 65-72.

[6]. P. Koehn, Statistical Machine Translation, Cambridge University Press, 2009.