DOI: 10.15439/2015F113
ACSIS, Vol. 5

Real-time Direct Translation System for Sinhala and Tamil Languages

Rajpirathap S, Sheeyam S, Umasuthan K, Amalraj Chelvarajah
Faculty of Information Technology, University of Moratuwa, Sri Lanka
Email: nova-fit10@googlegroups.com, amalraj@uom.lk

***Abstract*—Language barriers in day to day communication are common in all countries. In Sri Lanka we have a rising need for translation for Sinhala and Tamil to reduce language barriers and the statistical machine translation approach is more suitable for the concerned languages. Statistical machine translation method is one of the most promising and efficient method to perform machine translation for Sri Lankan languages likes Sinhala and Tamil. Statistical approach is more suitable for structurally dissimilar pairs of languages and efficient solution for large text translation. Sinhala and Tamil have a similarity in grammar and statistical approach will help to obtain more accurate results. We have developed a Real-time bi-directional translation system for both Tamil to Sinhala and Sinhala to Tamil for this research. We have used the Sri Lankan parliament corpus to train the language model. We have critically evaluated the both systems with parameter optimizations and have obtained the most accurate and efficient system. We have also utilized the scoring techniques like BLEU [2, 8] & NIST [2] for the system evaluation and we have integrated the MERT technique to tune the decoder.**

***Index terms*—Statistical machine translation, Natural language processing, Sinhala, Tamil, Machine Translation**

I. Introduction

A. Background

A*UTOMATIC machine translation* is one of the main concepts in simulating *Human Intelligence* which has several researches going on. The functionality of machine translation is to perform translation activities on natural languages which are complex and ambiguous. Machine translation comes under the learning area of Natural Language processing which depends on the subject areas like statistics, linguistics and computer science. Machine translation can be defined as *"Automatic translation from one language to another using computing devices and algorithms"*. A translation approach is about combining many techniques into one to get the translation right. Machine Translation can be divided into approaches such as transfer approach, Interlingua approach, direct approach and corpus based approaches which has two types like Example based and Statistical based)

B. The Statistical Approach

According to proven results statistical machine translation is one of the most efficient and effective translation approaches which is well matched with western and Indic languages. Statistical MT approach helps to finish up with mathematically easily decomposable model. It is not very complicated when compared to other rule based approaches and has been proven as the most promising approach to all-purpose text translation. SMT approach has the standard algorithms and models available which can be applied to any language pair, with large corpora with few linguistic assumptions. This help to minimize the development duration.

C. Goal, Objective, Scope & Motivation

The main goal of this research is to develop a real-time communication system which can perform statistical machine translation for Tamil and Sinhala.

Our objective is to research on the machine translation domain and create an efficient and accurate system than existing ones. The real-time system we have developed for this research performs machine translation in a very efficient way and can be used as a core for many types of software. Other than the development part of this system we also have some sections like MERT tuning, system evaluation using BLEU and NIST metrics.

The scope of this project is to develop a real-time instant text communication tool which can translate Sinhala and Tamil bi-directionally. The translation output is based on the type of the language corpora we use to implement the system. And the training data is updated continuous data from users through the reporting system which will increase the quality of the output with many users using it. First few months has been spent on research and learning of Natural language processing concepts. Few months period has been spent for the design and Implementation of the system. Software tools namely GIZA++, Moses, IRSTLM, MERT Module, NIST & BLEU module are the components that are built-in in the system.

The motivation to try out this project is to reduce the unavailability of translation systems which can be used for instant messages especially for Tamil and Sinhala. And previous researches in this domain are discontinued or not visibly used. Languages concerned in our researches are Tamil and Sinhala. When analyzing about the two languages the sentence structure on both languages seems to be identical in many occurrences. Disregard to some insignificant variations, the two languages has the same grammatical structure which is adequate to maintain the meaning understandable in both languages.

We are using Statistical Machine translation because it gives good results for even dissimilar language pairs and using rule based approaches for languages like Tamil & Sinhala will consume a lot time, effort and the result will be error full and inaccurate.

978-83-60810-66-8/$25.00©2015, IEEE

D. Statistical Machine Translation

Statistical translation systems work by learning how the grammar of source and target languages are defined. They begin to work with less number of dictionary entries and language resources. Developers can train the system by increasing the entries to handle complex and extensive translation scenarios. Google is one big player in SMT related NLP applications.

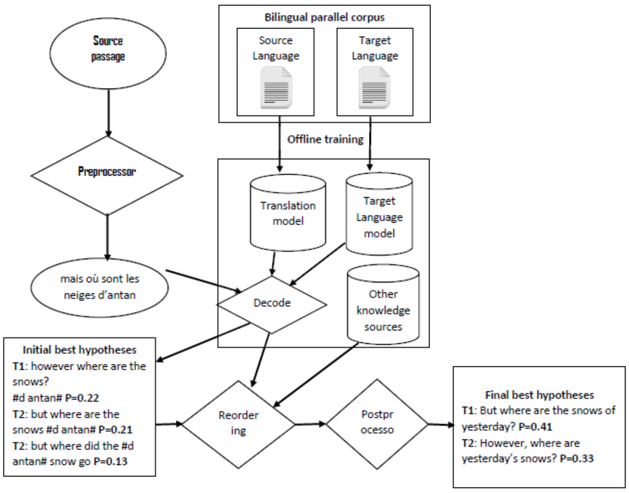
Source passage
Preprocessor
mais où sont les neiges d'antan
Bilingual parallel corpus
Source Language
Target Language
Offline training
Translation model
Target Language model
Decode
Other knowledge sources
Initial best hypotheses
T1: however where are the snows?
#d antan# P=0.22
T2: but where are the snows #d antan# P=0.21
T2: but where did the #d antan# snow go P=0.13
Reorder ing
Postpr ocesso
Final best hypotheses
T1: But where are the snows of yesterday? P=0.41
T2: However, where are yesterday's snows? P=0.33

Diagram 01 : Statistical Machine Traslation

Statistical MT systems usually work or get trained by breaking sentences into n-grams. Analyzing n-grams will improve accuracy and performance. A word may have many types of meanings but it will only have fewer meaning in a phrase form. Most of the SMT systems work on bi-gram and commonly tri-gram. Tri-gram simply means as three –word groups. Three-word groups are more than enough to process efficiently on the data set. Larger n-gram will require more power and time to analyze and translate. In a statistical machine translation system the common n-grams are tracked and more frequent translations are learned and used in the future translation activities. Statistical analysis of n-grams are happened which will analyze the position of the n-Gram with regard to the sentence. Mathematical translation models are updated after each and every translation which will result in accurate results. Faster processer will improve performance of training and tuning and reduce the time taken to it. Technologies such as C++, Java and Perl are used to develop NLP algorithms. As the languages are platform independent. They will enable their functionality on many Operating Systems. Training the SMT system extensively will lead to a highly accurate translation system. Compared to Rule based systems Statistical systems saves more time, money and effort. Rule-based systems for a language pair like Tamil & Sinhala will require many years and considerable amount of funding. Statistical systems can be trained to a good accuracy level in few months. This makes statistical approach less work intensive and that is the main reason we choose this approach for our short-term research purpose.

The SMT approach is more useful for corporate and government applications. One of the limitations in a Statistical machine translation approach is that it will require powerful computers for large training and accurate translation. The main concept in SMT systems is to select the target language phrase which has the maximum probability of being the translation on the source language phrase. A probabilistic model is a base for all the computations in the SMT system. Assumptions on Bayes Rules and Noisy channel model make the system less complex when generating the probabilistic model. When an input sentence is given into the system the target phrase with more probability will be selected as the output. This statistical translation approach can be briefly described in mathematical terms

$$\mathbf{t = argmax_t\ P\ (t/s)\ \ldots\ldots\ldots\ldots\ldots\ (1)}$$

Using Bayes' theorem, the value of P (t/s) can be given as

$$\mathbf{P\ (t/s) = (P\ (t)\ P(s/t))/\ (P(s))\ldots\ldots\ (2)}$$

Referring to equation (2), t can be written as,

$$\mathbf{t = argmax_t\ (P\ (t)\ P(s/t))/\ (P(s))\ldots\ (3)}$$

In equation (3), s is fixed in source language. As a result of this, P(s) can be removed from the equation when finding the sentence t. That has maximum probability. Then equation (3) becomes as equation (4)

$$\mathbf{t = argmax_t\ (P\ (t)\ P(s/t))\ldots\ldots\ldots\ldots\ (4)}$$

Element P (t) in equation 4 denotes the kind of sentences in Target language (T). P (t) is called Language Model of target language (T). The other element in equation 4, P (s/t) specifies how each sentence in target language (T) can be translated into source language sentences (s). This is called as the Translation Model. Equation (4) lays the foundation of statistical machine translation with the specification of two key components namely language model (P (t)) and translation model (P(s/t)).

Improving the language model in the SMT system is a manual job but we can use MERT techniques to train the system in minimum error rate which will result in accurate systems. MERT recommends a substitute training strategy for log-linear statistical translation models. MERT training is a straight forward method which will optimize translation quality using some automatic metric score. During the weight optimization of decoder parameters such as phrase translation table, language model, distortion, word penalty etc. MERT searches weight values that reduce translation errors. Metric scores supported by MERT are BLEU, NIST and TER. As the manual evaluation is hard and time consuming job, to make the evaluation more flexible metrics like BLEU and NIST were introduced. BLEU metric is a de facto standard which is used for Machine translation system evaluation. BLEU score gets the geometrical mean of modified precision score of the test corpora and multiply with some exponential brevity penalty factors. BLEU score increases to a higher value when the number of reference translations is increased. NIST [2] is a metric built on top of BLEU [2, 8] which enhances it with few modifications. One of the major modifications done by NIST is it assigns higher weights to rare or less occurring n-grams than regular ones. It simply applies the smoothing technique on rarely used n-grams.

II. Related Work

There are numerous researches and projects developed on Statistical machine translation in the recent past. Sri Lankan developers have worked on many Sinhala, Tamil related NLP systems in the past few years. In the reference [3], the developers had developed statistical machine translation systems for 4 European based languages such as English, German, Spanish and French. The data they used was from a general domain and large in size. German to English SMT systems has an average BLEU score of 0.236. Spanish to English SMT systems has an average BLEU score of 0.340. French to English SMT systems has an average BLEU score of 0.316. The research [1] is a SMT implementation for English and Sinhala Languages. This research includes several refinements. This research talks about MERT inclusion, Translation Model tuning, Language Model Tuning and various word alignment and reordering techniques. The BLEU score obtain at the end of the research was around 0.1500 which is a very low value.

The research [2], which was developed by UCSC, is one of the best references for SMT implementations. This SMT system was implemented for Sinhala and Tamil Languages. This research also uses the data of specific domain which is from various public websites. The best BLEU score obtained from this research for Sinhala to Tamil system was 0.185. Even though it's not a very successful research this paper talks about some viable techniques and approaches for SMT based applications. This research uses some old tools and this research was just the beginning of a successful SMT system. The further researches by this team have given great result on evaluation. This research is one main reference for our current research. After reading all the related work we learned every strength and weaknesses of the systems and have embedded that knowledge into our research to obtain higher scores. We have used newer and updated techniques to our current research and we have also contacted the previous researchers on this domain and brainstormed ideas with them to come up with a good final output. The innovative thing we are proposing through our research is using this traditional SMT concept and implementing them into a real time communication application. This application will be less complex and will support all kind of users facing language barriers during communication.

III. Data Preparation

In Sri Lanka when developing applications on Machine translation the only resource for datasets on Tamil and Sinhala is the parliament order papers on budget proceedings. We have used an electronic version of the parliament order papers which has parallel data on both Sinhala and Tamil. We used over 5000 phrases from each language which is totally more than 10000 sentences and more than 100000 words to train the system. One of the main reasons for the lack of development in translation systems in Sri Lanka is the unavailability of the corpus and the restrictions to obtain them. We obtained the Sinhala and Tamil Electronic version in a dirty manner where it is not aligned properly with the languages. The initial stages of the research was planned to develop a SMT systems in the chat domain for Sinhala & Tamil. After the research we found that parallel data Sinhala and Tamil in the chat domain is not available and building or creating one would require a lot of funding in the current situation. We limited ourselves to the parliament domain. The reason for using the parliament order papers is that the parliament members speak more formal languages which are not commonly used in the society. The languages are translated into more exact order in the parliament because in parliament discussion every word and phrase is important. We can 100% ensure that the meaning is not altered due to the variation in the language style. This process applies to both Sinhala and Tamil. One of the drawbacks of the system is that when we use a parliament corpus for the research the system is tuned to domain of parliament style translation so that we cannot use this system to translate non formal type of sentences or phrases. The obtained electronic versions of the parliament order papers were pre-processed before using them as inputs. Reasons for pre-processing are unwanted gaps between words, disordered words, disordering of Tamil Sentences aligned to Sinhala Sentences and some complex character are broken which makes it a garbage character. After cleaning the Sinhala and Tamil parallel data we can ensure that most of the sentences are in proper order in allowed manner, but still some issues will remain. We have to assume the pre-processed data as a perfect one. Redundant contents are removed randomly in some selected files and the final data set for the system is prepared. In a Statistical machine translation system implementation data preparation is essential and plays a major role in the final results. After the preparation of the datasets we did the selection process of the Data sets. We divided the data set into three such as

Training set, Tuning Set and Testing set. The table is given below. This table shows the number of word and phrases in the data sets.

TABLE 1: DATA SETS

Data Sets (Words and Phrases)							
Language Model		Training Set		Tuning Set		Testing Set	
Si	Ta	Si	Ta	Si	Ta	Sin	Ta
165k	77k	99k	78k	3425	3078	3110	3204
6550	6104	5887	5887	200	200	200	200

Addition to the above mentioned datasets, as a team we have prepared our own dataset which includes parallel sentences of normally used phrases in Sinhala and Tamil. We prepared more than 2500 parallel sentences and in our system we have added a feature where the user can report the admin with the translations. These translation sentences will be added to the system's training data and the SMT core will tuned frequently in a periodical manner. This will improve the output quality. One of the reasons we started to develop a real time SMT chat application is that we can create new and improved trained data using users' contributions. This is will lead to an improved data set creation which can be used for future projects related to SMT as well.

IV. OUR APPROACH

The system design is the heart of all system implementation. The design section consists of three main parts such as Architecture Design, Implementation Design and Evaluation Design. Architecture design is all about explaining how the modules and components are connected and working together. The implementation design illustrates about the implementation of folder structures and file paths which are used in the SMT system. The third one, evaluation design explains about how evaluation and experiments are carried out on the system to arrive at conclusions and results.

A. Architecture Design

The main Architecture of this Real-time Statistical Machine Translation system is basically not an innovative one or a new thing to be researched. It has a long standing research history in its hands. The main component of this project is the SMT module and we don't need to alter the entire design of the SMT but have to follow the best practices. SMT approach is language independent. Many of the SMT implementations follow the common architecture. In our research, the both systems use this uncomplicated architecture as its backbone. Rather than mixing all the modules together, we are defining the components explicitly and linking them according to the need. The SMT system we are developing for this research is a layered architecture. This architecture helps us to make the system uncomplicated and easy to understand. The layered architectures help to add and remove components.

In the design when considering about the data preparation component, the functionality of it is to obtain the whole data set and divide into subsets. These separate data sets are used for training, tuning and testing purposes, which will lead to efficient system evaluation. Addition to this we can do language modeling if a monolingual corpus is available. The whole data set is divided into two subsets. One is for language modeling and other one is for model evaluation. Model evaluation will not require huge data set. Before these steps the whole data set is tokenized. Formed outputs are the inputs to language modeling component. In case of Sinhala to Tamil system, Tamil language model should be produced. The un-tokenized whole data set is divided into three parts for training, testing and tuning. Training data set is the input of translation modeling component. Tuning data set is the input for the automatic tuning & decoding component and testing data set is the input for evaluation component.

The Language modeling module consists of all executable files created by IRSTLM. N-gram and n-gram count are two modules needed for LM modeling and evaluation respectively. Outputs of the language modeling component is the Language model and the perplexity scores. To obtain an optimal language model from evaluation, it is required execute the LM with various smoothing and discounting parameters and obtain the scores. Manual evaluation scores after each language modeling would help decide the optimal one. The next component is the translation model which is created using the training set. We have referred GIZA++ tool to implement this module which creates the word alignment and translation model. We can adjust the word alignment and reordering strategies to create or obtain the best Translation Model. Decoder configuration file is created at the end of this procedure so that the decoder script will start running by referring this script. The tuning component is responsible for changing decoder parameters & weights modifications which will result in change in evaluation metric scores like BLEU & NIST. The next component is the decoder executable. The inputs for this module are the test data set and the TM components such as phrase table and word alignment table and the decoder configuration file. The actual translation function happens inside this module. Source language file is translated and the translation process stops at the formation of translated output file. Prior to the automatic evaluation via three metrics, a little formatting is essential on the input data sets of metric-modules. Inputs of metric modules are translated output, reference text and source text. In Sinhala to Tamil system, reference file needs to be in Tamil which is the genuine translation of the Sinhala test data set. Scoring is restricted for only one reference file. Other than these above modules we have another two separate modules which are called as back up module and reload module. Back up module backups all system inputs, configurations and the outputs. Reload module reloads the backup data and configurations into the system. With all these components we have also having a simple client to client chat or communication application to fulfill the scope. This application uses the SMT module as the core and serves the users who want communicate in their native languages without any barriers.

B. Implementation Design

For a successful research a good Design is essential. As developers we have to also focus on implementation design to achieve what we wanted. Having good design architecture isn‘t lonely enough. Implementation design consist the details of folder and directory structure of the implementation which is very important for the performance and the efficiency of the system. A proper folder structure will also help for easy maintenance which will reduce errors and unwanted exceptions.

In our SMT system we have bi-directional implementation which includes Sinhala to Tamil and Tamil to Sinhala. Both the systems share some common files in the system. Some tools are configured and made common to both systems. Interaction with each system is feasible through bash scripts. In our system we will have two clients who does the communication part. The input string is received from one client end and then the SMT core translates it into a target language. The target string is sent to the client in the other side.

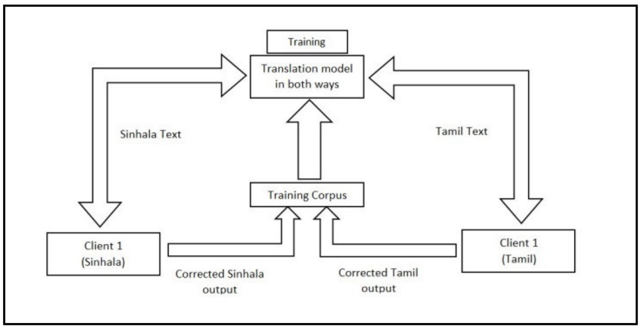

Diagram 02 : Workflow of our Final Translation Application

C. Evaluation strategy

Evaluation of systems is an essential part of this research. A plan or strategy is developed during the development phase about how we are going to evaluate the system. The evaluation strategies we have followed here are change parameters and algorithms in different combinations and obtain BLEU and NIST score from each system. From those results we can obtain an accurate and efficient system for the languages like Tamil and Sinhala. The evaluations results are presented in a statistical form. These combinations of evaluation end in many number of system backups and as we are dealing with many evaluation techniques we can call this as a good evaluation strategy.

V. IMPLEMENTATION

Major parts of our implementation are Sinhala to Tamil MT System and the Tamil to Sinhala MT System. For a successful research, excellent design architecture in hands endorses effective maintenance and help reducing mistakes. Having good design architecture isn't only enough. Implementation or the directory structure holds a place to handle executions and maintenance comprehensively. Good directory structures are formed bearing maintenance/ backup/restore simplicities in mind.

The **LM input files** directory will hold the Data sets (both Sinhala and Tamil) for language models creation and LM test files (both Sinhala and Tamil) are located here. Calculated perplexity values are written to files residing in "lm evaluation" directory. The **Input** directory will contain the inputs for both SIN-TAM and TAM-SIN system. In case of SIN-TAM, reference file (Tamil) should be the corresponding parallel corpus of test file (Sinhala). The **Output** directory will have the final de-tokenized translated output of the system. NIST, BLEU and TER scores are saved in the **Results** directory. The **Scripts** directory will contain some Bash scripts to control operations such as cleaning, backup, restore, and load backup, run components like Data model, language model, Decoder, Automatic tuning component and evaluation component of entire SMT. There is also a directory in our system called **Word** and this will store tool's output/input files.

Corpus has tokenized parallel corpora. Evaluation directory is with raw system output and formatted output files (for automatic metric evaluation). Language model is located inside "lm" directory. MERT files and tuned decoder configuration files are found in tuning directory. The same structure as that of SIN to TAM applies to TAM to SIN system also but instead of all Tamil files, Sinhala files should be there and vice versa. In backing up process, "input, output, work, lm evaluation, results and importantly all Configurations files and used commands" will be copied to a new location. This process is essential as lots of experiments need to be done and as outputs are required to be loaded again for further tests or for re-evaluation purposes.

The directories "IRSTLM, GIZA++, Moses, TER modules" have all installed files but when porting the system to another location these may be ignored as all executable are located in "executable". During our development we have used some modules from some tools such as IRSTLM, GIZA++ and Moses. The IRST Language Modeling Toolkit features algorithms and data structures suitable to estimate, store, and access very large n-gram language models. It can be used in Linux platform. IRSTLM offers the most advanced n-gram smoothing methods to estimate large LMs and approximated smoothing methods to estimate gigantic LMs. It includes methods for pruning and quantization of LMs, efficiently storing LMs on disk and in memory and it offers several language adaptation methods: linear interpolation, minimum discrimination information, and probabilistic latent semantic analysis.

Moses is one of the open source toolkits for statistical machine translation. Moses toolkit uses some tools for some of the tasks to avoid duplication such as GIZA++ for word alignments and IRSTLM for language modeling. The information about Moses was obtained from Moses documentation. The Moses toolkit has achieved some objectives such as Accessibility, Easy maintenance Flexibility, user-friendly, distributed team development and Portability. The Real-time chat application is developed using Java programming language. It is very light weight and we have developed that software in which it can be applied as a layer on top of the SMT core system and operate in good efficiency.

VI. Evaluation

A. Basic System Evaluation

Our evaluated system's entire language model is domain specific. And Number of 2-gram hit is relatively high in both language models. All n-gram hits in Sinhala are not as much as Tamil Language.

TABLE 2:
N-GRAM COUNTS FOR SINHALA & TAMIL

Language Model	Sinhala	Tamil
1-Gram	4703	5946
2-Gram	15337	16283
3-Gram	8474	13280

TABLE 3:
AUTOMATIC METRIC SCORES FOR BOTH SYSTEMS WITHOUT MERT TUNING

Systems	Sinhala - Tamil	Tamil – Sinhala
BLEU	0.4277	0.5599
NIST	4.4090	4.1244

After the evaluation we obtained highly excellent BLEU & NIST scores in both systems and Tamil to Sinhala System gives the highest from the two systems. The BLEU score results in high values which is a proof that the system has high accuracy. In the Sinhala to Tamil system, the value of BLEU is not that best as NIST. However all these metrics favors the output of Tamil to Sinhala system in a positive manner. The both systems gives high positive values on scoring other than the existing systems developed by the past researches and experiments.

B. Tuned system Evaluation

The good thing of the research was that we had very good score with normal settings, but more quality output can be expected after fine tuning the system. We used the Minimum Error Rate Tuning (MERT) technique to achieve more scores. After MERT [2] technique we received new scores. Below table shows the new values obtained for the both systems in a table format. The score values are in four decimal places. These scores show some great improvements.

TABLE 4:
AUTOMATIC METRIC SCORES FOR BOTH SYSTEMS WITH MERT TUNING

Systems	Sinhala - Tamil	Tamil - Sinhala
BLEU	0.5957	0.6693
NIST	4.4182	4.8563

It is very much acceptable that MERT has enhanced the scores of the both systems in a good rate. BLEU score has been improved in both systems by a good number. NIST score has been improved in the Tamil to Sinhala translation system but not much in the Sinhala to Tamil System. BLEU has been improved in the highest percentage and NIST get the least improvements. When talking about decision making of the system we have developed is that the scores are well improved values than previous research on SMT with local languages like Tamil and Sinhala. And we can mark this as a successful research. Addition to scores we have also calculated the time taken for the translations for both systems.

TABLE 5:
AVERAGE TIME TAKEN FOR TRANSLATIONS IN BOTH SYSTEMS

System	*1 - sentence*	*3-sentences*	*5- sentences*
Sin - Tam	0.778s	1.94s	2.052s
Tam - Sin	0.035s	0.249s	0.299s

VII. Conclusion

In a summary, both translation systems have given positive results and scores. One of the beneficial things from good BLEU scores is that we can extend the research to new heights with a positive system. Word alignment algorithm we used was *grow-diag-final* and reordering algorithm was *msd-bidirectional-fe*. According to the tables mentioned in the evaluation section the BLEU & NIST scores were better than existing system and it improved even more after MERT technique. The final BLEU scores of the achieved systems are 0.595668 & 0.669333 and also achieved NIST scores are 4.4182 & 4.8563. Both systems also have very less and efficient translation times. These systems can be improved into a highly accurate system by using large dataset of training data and changing the architecture.

VIII. Further Work

There are many future plans to improve the research and improve the system further more usable. One of the main further plans for this project is that to use a large dataset of Tamil & Sinhala phrase set to train the system. The data we used for this research for now is very minimal and with the real-time system we have developed with the reporting feature, we can hopefully prepare more amounts of translation pairs to create a translation with high quality. The limitation we faced while starting the project was the unavailability of chat data on Sinhala & Tamil. The human effort required to prepare the data is too high and costly, but we have started preparing the data using some volunteers and planning to take to the next level in the future. One step on that is the implementation of the automatic reporting system in the chat application. We will be working on the negative outcomes from this research. In the future we can differentiate the build architecture and make the translation module usable for application like Facebook and related technologies. One of our visions on this research is to make this translation functionality as a service and make it available for the developers to use it.

Acknowledgement

The authors of this research paper wish to acknowledge the supervisors and members of Faculty of Information Technology, University of Moratuwa who helped in many ways for the research. Special thanks to UCSC, Sri Lanka for the data and the past research papers related to machine translation. The authors have adapted few technologies and research procedures for this research from their past researches.

References

[1] J.U.Liyanapathirana, ―A Statistical Approach to English and Sinhala Translation, BSc. Thesis, University of Colombo School of Computing, Sri Lanka, July.

[2] R.Weerasinghe, ―A Statistical Machine Translation Approach to Sinhala- Tamil Language Translation.

[3] C.Callison-Burch, C. Fordyce, P. Koehn, C. Monz and J. Schroeder, ―Meta-Evaluation of Machine Translation‖, in Proc. 2nd Workshop on Statistical Machine Translation, 2007, p.136-158.

[4] Franz Josef Och, ―Minimum Error Rate Training in Statistical Machine Translation‖, in Association for Computational Linguistics.

[5] Doddington,G ―Automatic evaluation of machine translation quality using n-gram co-occurrence statistics". Proc. Human Language Technology Conference (HLT), 2002,p. 128—132

[6] R.Weerasinghe, ―A.R. bootstrapping the lexicon building process for machine translation between 'new' languages. In Proceedings of the Association of Machine Translation in the Americas Conference (AMTA), 2002.

[7] Och, F.J., Tillmann, C. and Ney, H. ―Improved alignment models for statistical machine translation. In Proceedings of the 4th Conference on Empirical Methods in Natural Language Processing (EMNLP), Maryland, 1999.

[8] K. Papineni, S. Roukos, T. Ward and W. Zhu, ―Bleu: a method for automatic evaluation of machine translation, in Proc. 40th annual meeting on association for computational linguistics -2002, 2002, pp. 311–318.

[9] P. Koehn, F. J. Och and D.Marcu, ―Statistical Phrase-Based Translation, in Proc. Joint Conference on Human Language Technologies and the Annual Meeting of the North American Chapter of the Association of Computational Linguistics,2002, pp. 1 – 7.

[10] Bernadette Varga, Alina Dia Trambitas-Miron, Andrei Roth, Anca Marginean, Radu Razvan Slavescu, Adrian Groza, ―LELA - A natural language processing system for Romanian tourism, in Proc. 4th International Workshop on Advances in Semantic Information Retrieval, 2014, pp. 281 – 288.

[11] Franz Josef Och, ―Minimum Error Rate Training in Statistical Machine Translation‖, in Association for Computational Linguistics, 2003, pp. 160-167.

[12] A. Birch, B. Cowan, C. Callison-Burch, M. Federico, N. Bertoldi, P. Koehn and H. Hoang, ―Moses: Open Source Toolkit for Statistical Machine Translation, in Proc. ACL 2007 Demo and Poster Sessions, 2007,pp. 177–180.