2019 National Information Technology Conference (NITC), 08-10 October, Colombo, Sri Lanka.

Evolutionary Algorithm for Sinhala to English Translation

A. Nugaliyadde
College of Science, Health, Engineering and Education
Murdoch University
Western Australia
a.nugaliyadde@murdoch.edu.au

J.K.Joseph
Department of Information Technology
Sri Lanka Institute of Information Technology
Sri Lanka
jenny.k@sliit.lk

W.M.T.Chathurika
Department of Information Technology
Sri Lanka Institute of Information Technology
Sri Lanka
thilini.w@sliit.lk

Y. Mallawarachchi
Department of Information Technology
Sri Lanka Institute of Information Technology
Sri Lanka
yashas.m@sliit.lk

Abstract— **Machine Translation (MT) is an area in natural language processing, which focuses on translating from one language to another. Many approaches ranging from statistical methods to deep learning approaches are used in order to achieve MT. However, these methods either require a large number of data or a clear understanding about the language. Sinhala language has less digital text which could be used to train a deep neural network. Furthermore, Sinhala has complex rules, and therefore, it is harder to create statistical rules in order to apply statistical methods in MT. This research focuses on Sinhala to English translation using an Evolutionary Algorithm (EA). EA is used to identifying the correct meaning of Sinhala text and to translate it into English. The Sinhala text is passed to identify the meaning in order to get the correct meaning of the sentence. With the use of the EA the translation is carried out. The translated text is passed on to grammatically correct the sentence. This has shown to achieve accurate results.**

Keywords— Machine Translation, Evolutionary Algorithm, Natural Language Processing

I. INTRODUCTION

With large amounts of context available in the internet in different languages, language translation has received higher attention from industrial and research perspectives, and in order to handle large stacks of text data, Machine Translation (MT) was introduced. MT targets the usage computational models in order to achieve meaningful translations from one language to another.

In order to develop an MT computational model, it is required to understand both languages, or to have language specialists. Therefore, it is not always practical to develop computational models for MT. Computational MT models have evolved from simple rule based models to deep learning models [1]. Statistical models have shown to perform well in many natural language understanding tasks [2] [3]. Applying statistics requires a clear understanding of the language and the language structure [4]. However, it can be hard to achieve a sound knowledge on languages which are not popular, or which have complex structures [5].

Compared to other popular languages, Sinhala language has a smaller digital footprint. Therefore, using Sinhala for natural language processing tasks is a difficult task [2]. Furthermore,

Sinhala has a complex language structure, and therefore it requires an expert knowledge on the language [6]. Therefore, applying computational models to Sinhala language can be a hard task.

Natural language processing tasks for Sinhala language have been conducted using many techniques. The rule based approach was used in simple Sinhala natural language tasks [7]. Similarly MT for Sinhala was also done using rule based approaches [8, 9]. However, since language is a complex thing, statistical models were created and tested for Sinhala. Statistical models were used for Sinhala natural language tasks. The statistical models show a potential in achieving better natural language models for Sinhala. Statistical models used for MT in Sinhala have gained traction [10] [11]. Statistical models have been vastly used to create complex Sinhala language models, and these models were used on many industrial language tasks since the models are better. However, statistical models fail to capture all features of language models [12]. Statistical models require hand selected features, and therefore, expert knowledge is required [13] [14].

Deep learning does not require any feature selections, but it is capable of achieving higher accurate results compared to statistical methods and rule based models [12, 15-17]. Deep learning has been applied for MT [18] [19]. A deep learning model, however, requires larger dataset to learn, and this requirement to use large data enables the deep learning model to generalize [20] [21]. Languages with smaller digital data presence fail in producing better results using deep learning [7]. Therefore, applying deep learning for MT in Sinhala does not have a high potential for achieving comparative results.

In this paper, an Evolutionary Algorithm (EA) is used for MT from Sinhala to English. EA is an algorithm which focuses on the evolutionary mechanism. EA has 4 steps: Initialization, selection, genetic operation and termination. These steps are similar to natural selection in the environment. Until final batch is created, selection and genetic operation iterates [22]. EA has the possibility of achieving natural language tasks [23]. This is

978-1-7281-5569-2/19/$31.00 ©2019 IEEE

one of the first studies that have used EA to achieve MT. Our model uses EA in order to translate the text and generate the final text using the evolution. Direct translation would generate the English text. EA would use the evolutionary iteration to align and correct the grammatical structure. This would generate the final proper meaningful translation. EA does not require a large training dataset enabling languages such as Sinhala to be used in MT.

II. RESEARCH METHODOLOGY

A language model is a network containing words or the alphabet, and it is used for the training purpose. Also this model holds the vocabulary of the language. That was created by using Cygwin software in order to generate binary format of Language model.

A. Meaning Identification

Language translation is entwined with “Meaning Identification", and the base of translation is the identification of appropriate meaning of words with several different meanings or the word sense disambiguation [1]. According to the rest of the content of a sentence, the meaning of a word will differ. The hardest part is to identify the correct meaning of the word by considering the remaining parts of the sentence. Most of the researches have failed due to the incorrect solutions of this meaning identification function. Therefore this function is well known as a crucial one, which will always need the most attention. In this research, the Sinhala text will be the input. Therefore we have to identify the meaning of a Sinhala text which is even harder than identifying the meaning of English texts. So far, there is a number of researches that have been conducted for the English language. As a result they have found so many different ways to get the meaning of the text such as taggers, parsers etc. [8].

There are few researches conducted for the meaning identification in language translation for Sinhala. But the main problem that can be identified is that these researches have aimed at word by word translation. Although the researches aimed about word by word translation, and to get an accurate result, we have to consider the whole sentence and identify the meaning. When doing this meaning identification by considering the whole text instead of only one word, we get the relevant word and then we have to consider about the remaining words of the sentence. Then we can get the meaning of the text as a whole, which will lead to an accurate result [9]. As an example, if we have four words in a sentence and we want to get the meaning by considering one word at a time there will be a word by word translation. In that case, we do not bother regarding the collective meaning of the words and we consider about single words. But this will lead to lot of inaccurate results because the same word along with few other words can give a different meaning. In order to perform the meaning identification task successfully, the concept called Point-wise Mutual Information (PMI) will be used.

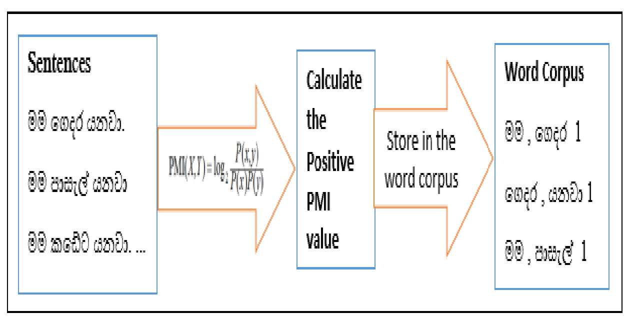
By using PMI values, we can get the most occurred meanings for a given word and thereby we can avoid the meanings which will never be there. The concept of the PMI is, when there are two words called x, y and if they have P(x) and P(y) the mutual information will be calculated by using the following formula.

$$\mathrm{PMI}(X,Y) = \log_2 \frac{P(x,y)}{P(x)P(y)}$$

The main idea of the PMI value is to consider the probability of the x and y together along with the probabilities of the x and y separately. The answer of the above formula will be considered and if it is a high value, it says that there is truly an association between x and y [13]. Furthermore, the values from the PMI will be taken as Positive Point-wise Mutual Information (PPMI) by replacing the PMI values that are negative values with zero [11]. By that we can avoid the unnecessary mistakes of language translation. This research was conducted in the Windows environment by using Python as the language. Python is a mostly used high level language which supports multiple programming paradigms. NLTK is a free and open source platform for Python programs which helps to work with human languages.

A word corpus is being built in the memory by calculating the relevant PPMI values. Then in the meaning identification process, this word corpus is used as an aid. In this word corpus, there are texts along with PPMI values for each text. Therefore when a text is given as an input for the meaning identification function, the word corpus that we have built is used to identify the meanings by considering the whole text and at the same time we can avoid unnecessary mistakes as we can identify the meanings that can never occur with the rest of the content of a text. This will lead to get an accurate result in this meaning identification function.

Fig. 1. Meaning Identification using PPMI



B. Translation Module

For the translation phase, a meaning identified Sinhala sentence or a paragraph is an input. The system splits sentences and does the tokenization, it leads to do dictionary lookup and lexical analysis using Sinhala to English dictionary. The system needs to catch unknown words, digits, signs etc. Then the system will morphologically analyze the tokens and identify the subject-predicate of the sentence. After that, the system will synthesise and morphologically process the target language (English). Finally, the sentence is rearranged to subject-verb-object format using POS tagging (Part Of Speech

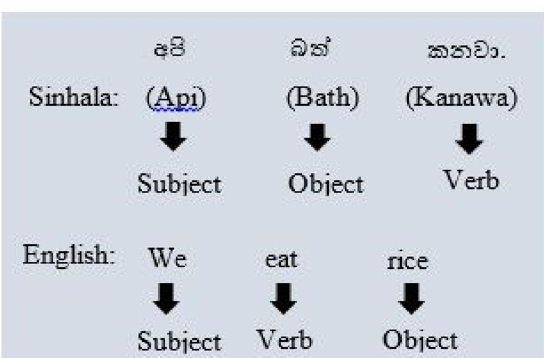
Tagging) [1]. Discussing about the direct or literal translation process, it is a dictionary based machine translation, which is based on entries of the language dictionary. The first generation of machine translation was entirely based on machine readable or electronic dictionaries. This methods is still helpful in translation [12].

The final output of the direct translation phase is not a complete intermediary sentence structure. A word-for-word (literal) translation occurs in this phase. Translation proceeds in a number of steps, each step dedicated to a specific task. The most important component is the bilingual dictionary. In this method, the Source Language text is structurally analyzed up to the morphological level, and matched for a particular pair of source and target language. This process of this system depends on the quality and quantity of the source-target language dictionaries, In order to achieve a higher accuracy level, "Arutha" Dictionary used about seventy two thousand distinct words.

Commonly Machine Translation systems are bilingual (designed for two particular languages). But there are multilingual systems as well (designed for more than two languages). A bilingual systems can be either unidirectional or bidirectional. Usually majority of bilingual systems are unidirectional, and they translate from one Source Language (SL) to one Target Language (TL). Multilingual systems are always bidirectional.

Ambiguity of a natural language is a major barrier for developing a quality translator. There are two types of ambiguity that occur in translating called Structural ambiguity and Lexical ambiguity [10]. The grammatical rules of Sinhala and English languages are different. In English language the sentence formation follows Subject Verb Object (SVO) whereas in Sinhala the sentence formation is in the form of Subject Object Verb (SOV). Differences of Sinhala and English sentence formation patterns cause grammatical errors when it comes to direct translation.

Fig. 2. Sentence format of Sinhala and English



To create a dictionary of Sinhala and English fonts in the system we have to go through the following process.

Download the appropriate Sinhala fonts in your system then copy .ttf file and paste the file in the **FONTS** folder which is available in the **Control Panel** of the PC**.** There are two ways to type the Sinhala text. First method can be used by the people who know the keyboard format of Sinhala fonts. After adding the fonts we need to go to **Control Panel Region – Languages preferences,** add Sinhala and okay it. The following method is a simple method to type Sinhala.

We need to type the text in English which will directly convert the text into Sinhala. To achieve this process we need to download Google input tool. We need to create Sinhala words and their equivalent English words text dictionary.

A meaning identified Sinhala sentence or a paragraph coming from the meaning identification function is an input for the direct translation function. After receiving the text, the sentence is separated into words based on the delimiters (space, commas, full stops etc.). The separated words are stored in an Array List. From the Array List each and every word will be checked with the dictionary file and they will be translated into English directly.

C. Evolutionary Algorithm & Grammar

The aim of the research is to translate Sinhala to English through meaning identification by using evolutionary algorithm, which is used to get grammatically correct sentences. EA uses selection and variation as basic principles. Competition among living beings is represented by selection, to survive and reproduce their genetic information some should be better than others, and natural selection is represented by this process. Each solution is given a change to reproduce and the quality is assessed by the fitness factor of each context. Variation shows the capacity of creating new living beings by using mutation. Likewise words will be selected and varied to create best quality sentences using Evolutionary Algorithm.

```
BEGIN
        INITIALISE population with unordered words;
        EVALUATE the candidate;
        REPEAT UNTIL (TERMINATION CONDITION is satisfied) DO
                SELECT parent;
                EVALUATE the parent;
                MUTATE by randomly pasting words;
                EVALUATE each child;
                SELECT fittest individual for next generation;
        OD
END
```

The common idea about the evolutionary algorithm is the same. When a population is given, only the fittest in the environment will survive, and it results in the increase of the fittest ones of the population and in the increase of the quality of the population. The fitness function is a requirement to fulfill the adaptation to the system. It helps to get an accurate result, and therefore, it is a measure of improvement. Based on the fitness of the sentence, some better sentences are chosen to seed the next generation by using mutation factor. Mutation factor randomly pastes words in order to gain new sentences. This process can be iterated until a sufficiently fit and grammatically fluent sentence is created. Variation operation creates the necessary diversity, and selection acts as a force that improves quality. The combination of variance and selection improves the fitness value of the sentence. It is good to see a optimized value or at least an approximate value by approaching optimal values closer and closer over its process. Every time the process runs, it makes the population adapt to the environment better.

NP: {<PRP>?<JJ.*>*<NN.*>+}
CP: {<JJR|JJS>}
VERB: {<VB.*>}
THAN: {<IN>}
COMP:
{<DT>?<NP><RB>?<VERB><DT>?<CP><THAN><DT>?<NP>}

As mentioned above, this process will create much accurate result, and it begins after identifying the grammar of that particular sentence. Firstly, grammatically incorrect sentence is taken into a list and by using nltk libraries, the sentence is broken into words, which are tagged as verbs, nouns etc. By using nltk library, a key and a value are given to each sentence in order to identify the words separately. By using the key, we can access the words, and it helps when placing the words in correct grammatical form. Then the created context is transferred to the algorithm in order to check fitness. In fitness function, the distance between target and the candidate sentences is calculated to check whether it is close to the target. If it is equal to zero, the expected output has been reached, or else, a sentence which is closest to the expected target is taken and mutated. A random number is created and that number is used as the distance between the two words, and words are randomly moved according to that number. 100 words are created at once, and this process is goes on until (fitness < 0 and count < 1000) the condition is fulfilled. By setting the count's maximum value to 1000, CPU utilization is limited; otherwise some processes might go up to infinite times. Then the fitness of created sentences is checked, the highest fitness is taken, and it considered as the parent of the next generation. This process is iterated until it reaches zero.

III. CONCLUSION

This paper introduces a model which uses EA to apply for MT. The proposed model translates English to Sinhala. However, a similar model can be applied to languages which have minimal digital text. The model directly translates Sinhala text into English. The direct translation would not hold the correct meaning of the Sinhala text. Using EA, the translated text is converted into a meaningful English text. Evolutionary process would continue until the text is grammatically correct. Furthermore, this study shows the possibility of applying EA to languages which have less digital text. Therefore, without moving towards deep learning, it is possible to achieve MT without training data, with high accuracy.

REFERENCES

[1] J. Hirschberg and C. D. Manning, "Advances in natural language processing," Science, vol. 349, pp. 261-266, 2015

[2] K. Senevirathne, N. Attanayake, A. Dhananjanie, W. Weragoda, A. Nugaliyadde, and S. Thelijjagoda, "Conditional Random Fields based named entity recognition for sinhala," in 2015 IEEE 10th International Conference on Industrial and Information Systems (ICIIS), 2015, pp. 302-307.

[3] E. Cambria and B. White, "Jumping NLP curves: a review of natural language processing research [review article]," IEEE Computational Intelligence Magazine, vol. 9, pp. 48-57, 2014.

[4] C. D. Manning, C. D. Manning, and H. Schütze, Foundations of statistical natural language processing: MIT press, 1999.

[5] J. Jayakody, T. Gamlath, W. Lasantha, K. Premachandra, A. Nugaliyadde, and Y. Mallawarachchi, "“Mahoshadha”, the Sinhala Tagged Corpus Based Question Answering System," in Proceedings of First International Conference on Information and Communication Technology for Intelligent Systems: Volume 1, 2016, pp. 313-322.

[6] B. Hettige and A. S. Karunananda, "A Morphological analyzer to enable English to Sinhala Machine Translation," in 2006 International Conference on Information and Automation, 2006, pp. 21-26.

[7] P. Antony, "Machine translation approaches and survey for Indian languages," International Journal of Computational Linguistics & Chinese Language Processing, Volume 18, Number 1, March 2013, vol. 18, 2013.

[8] B. Hettige and A. Karunananda, "Developing lexicon databases for English to Sinhala machine translation," in 2007 International Conference on Industrial and Information Systems, 2007, pp. 215-220.

[9] Y. I. Samantha Thelijjagodzf and T. Ikeda, "Japanese-Sinhalese “machine translation system Jaw/Sinhalese," Journal of the National Science Foundation of Sri Lanka, vol. 35, p. 2, 2007.

[10] R. Weerasinghe, "A statistical machine translation approach to sinhala-tamil language translation," Towards an ICT enabled Society, p. 136, 2003.

[11] S. Sripirakas, A. Weerasinghe, and D. L. Herath, "Statistical machine translation of systems for Sinhala-Tamil," in 2010 International Conference on Advances in ICT for Emerging Regions (ICTer), 2010, pp. 62-68.

[12] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," Nature, vol. 521, pp. 436-444, 2015.

[13] A. Nugaliyadde, K. W. Wong, F. Sohel, and H. Xie, "Reinforced memory network for question answering," in International Conference on Neural Information Processing, 2017, pp. 482-490.

[14] F. Farhath, S. Ranathunga, S. Jayasena, and G. Dias, "Integration of Bilingual Lists for Domain-Specific Statistical Machine Translation for Sinhala-Tamil," in 2018 Moratuwa Engineering Research Conference (MERCon), 2018, pp. 538-543.

[15] J. Weston, S. Chopra, and A. Bordes, "Memory networks," arXiv preprint arXiv:1410.3916, 2014.

[16] M. Hu, Y. Peng, and X. Qiu, "Mnemonic Reader: Machine Comprehension with Iterative Aligning and Multi-hop Answer Pointing," 2017.

[17] B. Peng, Z. Lu, H. Li, and K.-F. Wong, "Towards neural network-based reasoning," arXiv preprint arXiv:1508.05508, 2015.

[18] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, et al., "Learning phrase representations using RNN encoder-decoder for statistical machine translation," arXiv preprint arXiv:1406.1078, 2014.

[19] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," arXiv preprint arXiv:1409.0473, 2014.

[20] A. Nugaliyadde, K. W. Wong, F. Sohel, and H. Xie, "Language Modeling through Long Term Memory Network," arXiv preprint arXiv:1904.08936, 2019.

[21] S. Boukoros, A. Nugaliyadde, A. Marnerides, C. Vassilakis, P. Koutsakis, and K. W. Wong, "Modeling server workloads for campus email traffic using recurrent neural networks," in International Conference on Neural Information Processing, 2017, pp. 57-66.

[22] C. A. C. Coello, G. B. Lamont, and D. A. Van Veldhuizen, Evolutionary algorithms for solving multi-objective problems vol. 5: Springer, 2007.

[23] L. Araujo, "Part-of-speech tagging with evolutionary algorithms," in International Conference on Intelligent Text Processing and Computational Linguistics, 2002, pp. 230-239.