Improvement on Low Resources Machine Translation: English-Sinhala

Ziyuan Lin[1], **Zihao Zhou**[2], **Sanshan Guo**[3]

[1]ziyuanli@andrew.cmu.edu [2]zihaozho@andrew.cmu.edu [3]@sanshang@andrew.cmu.edu;

Abstract. *A large amount of language pairs are under low resource condition due to the insufficient official parallel data. Among all low resource language pairs, we selected English-Sinhala as our study target because of current machine translation model's bad performance on it. In this assignment, we applied some methods to improve the BLEU score of English-Sinhala translation, including data cleaning, adding domain tags and multilingual training. Then we dug into a specific problem we ran into, made assumption and set up some experiments to prove it.*

Key Words*: low resource, Sinhala, machine translation*

Github Repository*: https://github.com/andy-lzy/11731-project*

1. Introduction

1.1. Related Work

In Francisco et al's work[1], supervised, unsupervised, semi-supervised and weakly supervised machine translation models are trained on Sinhala-English. In a word, English to Sinhala model generally have a better performance, for example, in the supervised model, English to Sinhala model's BLEU score is 7.2, while the reversed model only gets 1.2. It's puzzling that in this paper, all models are trained based on a data set containing more than 500 thousand parallel sentences. Judging by the data size, the model's BLEU score is unexpectedly low.

1.2. Data Description

We use the same data set in the Francisco et al's work. The training data set is a combination of GNOME Ubuntu handbook and language sentences from OpenSubtitles[2], and the test and development data are same to Francisco et al's work [1]. After taking a look at the training data and validation data, we can easily find a mismatch of topics between the two data sets.

1.3. What We Want to Do

As we have stated in the 1.1, a supervised machine translation model gets a very low BLEU score on this Sinhala-English language pair. We want to analyze why the score is that low and find out ways of improving the BLEU scores. Also, we noticed that the Sinhala-English and English-Sinhala translation has a distinguishable BLEU score difference. Therefore, we want to explore both directions and apprehend the unexpected score difference.

2. Methodology and Results Analysis

We trained two model structure in this assignment. The first model structure is the model with the same structure with the model we made in assignment two and we call it 'assignment2 model' here. Another model is the model structure same to [1], named 'Flores model' in this report. The resaon why we trained two model is that we have analyzed the assignment2 model code by code and were very familiar of the structure of whole model, which makes it easy for us to tune and modify. On the other hand, Flores was using $fairseq$[3] for model training, which made it much harder for us to modify the structure of model.

	$baseline$	$clean$	$wiki$	$clean + wiki$	$clean + wiki(etag)$	$clean + wiki(dtag)$
en-si	0.23	0.26	0.88	1.30	**1.32**	1.31
si-en	1.90	1.54	2.12	2.90	2.93	**3.15**

Table 1. Results of using en-si dataset to train assignment2 model

2.1. Clean original data and make it more suitable for training

In the benchmark of the supervised machine translation of Flores Evaluation datasets, the training data used for train from English to Sinhala is the combination of OpenSubtitles[2] with 601K parallel sentences and 3.6M tokens and GNOME/KDE/Ubuntu corpus with 46K parallel sentences and 151K tokens. However, after we roughly looked through the dataset, we thought these training data is kind of noisy and not suitable for training the NMT model directly. For example, some Sinhala sentences have a lot of repeated punctuation or symbols, which cannot be found in the parallel English sentences, and some Sinhala sentences contain normal English words that are not translated, which does not make sense either.

Thus, the first thing we need to do is removing those noisy sentences and make it suitable for training. According to the specific characteristics of these data, we specially write filter code for these data to adapt to these characteristics and delete all parallel sentence violating the rule of selection strictly.

For translation from English to Sinhala, as can be shown in the first and the second column in table 1 in the first row, after cleaning the data, the BLEU score increased from 0.23 to 0.26. That is to say, cleaning up data based on the specific characteristics of data is a way to lead to better performance on translation.

However, for translation from English to Sinhala, shown in the first and the second column in table 1 in the second row, the BLEU score decreased after cleaning the data. We have not trained this model until we found out after cleaning training dataset, the BLEU score of translating from English to Sinhala trained by Flores model was much lower than the model trained on uncleaned data. Based on this result, we explored further analysis and experiments in section 3.

2.2. Increase training data with full-domain data

After the analysis, we found that the domain of the data used in formal training is mostly on the religious domain while the domain of test and validation set is from parallel sentences in Wikipedia. We decided to train the assignment2 model with data in

WikiMatrix[4], which contains about 90K parallel sentences. The BLEU score of training model from English to Sinhala on only wikiMatrix data and the combination of cleaned data and WikiMatrix data is 0.88 and 1.30 respectively, and from Sinhala to English is 2.12 and 2.90, which makes a huge improvement compared with the former results.

2.3. Add tags on either encoder or decoder side for different domain

From the method mentioned above, we found that domain is a very important attribute. Therefore, based on previous methods, we added tags to either encoder or decoder to investigate whether it can make improvement and compare the two methods to see which one is more effective.

While adding tags on decoder side, in order to avoid adding a new word for each target sentence, we replaced the $< sos >$ token for different tags. We used same strategy when adding tags on encoder side.

The BLEU score of adding tags on encoder and decoder side when translating from English to Sinhala is 1.32 and 1.31 respectively. The result of adding tags on encoder side is slightly better than on decoder side. When translating from Sinhala to English, the BLEU score also increased after adding tags to different domain, but this time the BLEU score we got by adding tags on decoder side was a little higher than adding tags on encoder side. Since the BLEU score is really low, it is hard to draw any conclusion only based on this specific situation, but we can say that adding different tags on in-domain data and out-of-domain data is likely to improve the performance of translation on in-domain sentences.

In Francisco et al's work [1], their model is more advanced than ours, which got a BLEU score of 1.2 translating from English to Sinhala and a BLEU score of 7.2 translating backwards. However, through a series of methods we mentioned above, we got a higher BLEU score of 1.32 with a not so well-performed model. Similarly, there is also an improvement from Sinhala to English, which also shows from the side that our methods can greatly improve the performance of translation.

2.4. Comparison between two models

In order to verify the efficiency of our methods and prove the validity of the analysis we have made, we used our new produced datasets to train the Flores model. Because of the difficulties of modifying their model, we avoided using tag-policy.

	$baseline$	$clean$
en-si	1.2	**1.46**
si-en	**7.2**	5.09

Table 2. Training Flores model on cleaned dataset

We trained the Flores model with cleaned dataset, the result is shown in table 2. It is clear that only cleaning the data can get a better BLEU score for English-Sinhala than baseline, but on the contrary, the BLEU score for Sinhala-English is much lower than baseline after data-cleaning. That is a strange but interesting situation, further experiments and analysis will be metioned in section 3.

2.5. Multilingual training with related languages

1. **Dhivehi**.
   The Maldivian language, whose endonym is Dhivehi, is an Indo-Aryan language spoken in the South Asian island country of the Maldives. [5]. According to Wikipedia, the closest relative of Sinhala is the Maldivian language[6]. The genetic proximity[1] between Dhivehi and Sinhalese is 48.6, which means these two languages are closely related[7].
   We built a multilingual corpus by combining original cleaned Sinhala-English parallel data and Dhivehi-English parallel data, added tags on decoder side and trained a assignment2 model. However, as shown in the first column in table1, the BlEU score is even lower than the baseline. After checking the Dhivehi-English dataset we were able to find, the domain is also on religious domain, which is very limited and result in a worse performance on translation.

	baseline	en-(si,dv)	en-(si,hi)
en-si	0.23	0.20	**1.81**

Table 3. Multilingual training with decoder tag

2. **Hindi**.
   After realizing the importance of domain, we started trying to find another related language that have parallel dataset with English in Wikipedia domain.Hindi is an Indo-Aryan language spoken in India and across the Indian subcontinent and it is the third most-spoken language in the world[7]. Therefore, Hindi has a really large parallel corpus with multiple domains. However, the genetic proximity between Hindi and Sinhalese is 57.3, which means those two languages are related, but not so much. We've decided to find out whether an in-domain dataset of a less related language can improve the performance of translation.
   We used the same method mentioned above. The BLEU score we got was 1.81, which is the highest BLEU score in this report using assignment2 model. Which means not-so-related language with in-domain dataset can improve the performance of translation. Compared with the baseline BLEU score of 0.23 in assignment2 model and 1.2 in Francisco et al's work, we've made a great improvement.

3. Further Analysis

As mentioned above, after cleaning the raw dataset, we got a better BLEU score when translating from English to Sinhala on both assignment2 model and Flores model, however, on the contrary, when translating from Sinhala to English, the BLEU score dropped on both models. This strange phenomenon caught our attention, and we decided to focus on explaining this situation in the remaining project.

3.1. Assumption

Firstly, considering that the difference of performances under different datasets and different languages is similar when training on those two models, we think the most important factor is possibly derived from the dataset. Then we deeply looked into the raw

[1]Genetic proximity. URL: http://elinguistics.net/Compare_Languages.aspx

training data. One important feature of the data we've found was most of noisy sentences were Sinhala sentences, while the parallel sentences in English seemed good. Intuitively, that also makes sense because there are a large amount of sentences are from bible or Ubuntu, those sentences are more likely translated from English to Sinhala, which makes English sentences remain clean.

If we assume most of the noisy parallel data is only noisy on Sinhala side, the phenomenon we've mentioned above can be perfectly explained. When we are translating from English to Sinhala, if Sinhala sentences are noisy, the noisy sentences are on the target side, which will cause bad influence on the decoder and make the translation less natural. After we removing those noisy sentences, we kind of removed the bad influence and improved the performance of the model. However, when we are translation from Sinhala to English, the sentences on target side are pretty good, the noisy Sinhala sentences on source side will not have too much bad influence on the model. On the contrary, since we only have about 640,000 raw sentences and 430,000 cleaned sentences for training, adding back about 110,000 sentences we've removed by cleaning, which have little noise on target side, can totally improve the performance of decoder. We can also consider those noisy sentences as good back-translated English-Sinhala parallel dataset with some noise. That can explain why training on the raw data can get a Sinhala-English model with better performance than training on the cleaned data.

3.2. Experiment

To prove our conjecture, we tried to find out how those noisy sentences will influence the decoder on both directions. We trained language models for both English and Sinhala based on two different datasets(raw & cleaned), and use the same validation and test dataset talked above to test the performance of the models.

We used KenLM [8] to train the language model. We've trained 2 language models based on raw dataset and cleaned dataset on each side of language. The score we've used to evaluate the model is mean log10 probability, which is basicly the sum of log10 probability for each sentence divided by the number of sentences. The result is shown below.

	dev	test
raw	-57.75	-71.20
cleaned	**-52.27**	**-66.62**

Table 4. Sinhala

	dev	test
raw	**-61.08**	**-67.89**
cleaned	-61.66	-68.08

Table 5. English

Since the score is the log10 possibility to generate that sentence, a higher score denotes a better performance of language model. In table 4, it is clear that the score of Sinhala model trained on cleaned data is much higher than the model trained on raw data on both validation and test datasets. However, table 5 shows that cleaning data cannot get a better English language model, the model trained on raw data gets higher scores on both validation and test datasets.

The result of this experiment totally fits the assumption we've made, which can prove that the noisy sentences on the Sinhala side could be the main factor that causes this phenomenon.

We ran another experiment to further prove our assumption. We combined the

WikiMatrix dataset with either cleaned dataset and raw dataset, and trained Flores model on these two datasets seperately. The result is shown in table 6. It clearly shows that adding WikiMatrix dataset has a great influence on the translation. It also shows that the model trained on WikiMatrix and raw dataset is better than the model trained on WikiMatrix and cleaned dataset, which further proves using the raw data can gain a better performance of translation than using cleaned data when translating from Sinhala to English.

	$baseline$	$wiki + clean$	$wiki + raw$
si-en	7.2	9.55	**9.84**

Table 6. Multilingual training with decoder tag

4. Conclusion

In this project, we try different methods to improve the performance of translation.

We trained a new translation model based on the model in assignment 2 and compared it with the Flores model. Before training, we cleaned the data set by filtering noisy sentences with our selection rules. Then, we increased full-domain data to training data for the purpose of matching training and validation data domains. Also, we added tags on either encoder or decoder, and the best model we have got was the ones that were using domain tag. In addition, we implemented multilingual training with related languages in different domains, which further improved the performance of model. As a consequence, we optimized the translation in both directions.

We also dug into the problem that cleaning data improves the performance of translating from English to Sinhala while deteriorating the performance of Sinhala to English. We have run several experiments to prove our assumption. Which turned out that the main factor was most of the noise exists only on the Sinhala side, and the unnaturalism in target sentences will cause bad influence on the performance of translatoin.

References

[1] Francisco Guzmán, Peng-Jen Chen, Myle Ott, Juan Pino, Guillaume Lample, Philipp Koehn, Vishrav Chaudhary, and Marc'Aurelio Ranzato. Two new evaluation datasets for low-resource machine translation: Nepali-english and sinhala-english. *CoRR*, abs/1902.01382, 2019.

[2] Pierre Lison, Jörg Tiedemann, and Milen Kouylekov. Opensubtitles2018: Statistical rescoring of sentence alignments in large, noisy parallel corpora. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, 2018.

[3] Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*, 2019.

[4] Holger Schwenk, Vishrav Chaudhary, Shuo Sun, Hongyu Gong, and Francisco Guzmán. Wikimatrix: Mining 135m parallel sentences in 1620 language pairs from wikipedia, 2019.

[5] Wikipedia contributors. Maldivian language — Wikipedia, the free encyclopedia, 2019. [Online; accessed 10-December-2019].

[6] Wikipedia contributors. Sinhala language — Wikipedia, the free encyclopedia, 2019. [Online; accessed 10-December-2019].

[7] Wikipedia contributors. Hindi — Wikipedia, the free encyclopedia, 2019. [Online; accessed 10-December-2019].

[8] Kenneth Heafield. Kenlm: Faster and smaller language model queries. In *Proceedings of the sixth workshop on statistical machine translation*, pages 187–197. Association for Computational Linguistics, 2011.