

DHOLUO –SWAHILI MACHINE TRANSLATOR USING OPENNMT –PY

Peter Kimanga

Dr. Benson Kituku

Department of Computer Science

Department of Computer Science

Dedan Kimathi University of Technology

Dedan Kimathi University of Technology

Email:

Email: benson.kituku@dkut.ac.ke

peter.kimanga23@students.dkut.ac.ke

Abstract

Machine Translation (MT) has seen significant advancements, particularly with the rise of Neural Machine Translation (NMT), which leverages neural networks to produce more fluent and contextually accurate translations. This research aimed to develop a specialized NMT model to translate texts from Dholuo to Swahili. The model was implemented using the OpenNMT-py toolkit, a powerful and flexible open-source framework for neural machine translation. It utilized the Kencorpus dataset, which contains texts in both Dholuo and Swahili. This dataset underwent rigorous cleaning processes to remove noise and inconsistencies, ensuring that the data used for training the model was of high quality. The developed NMT model was evaluated using the BLEU (Bilingual Evaluation Understudy) score, a standard metric for assessing translation quality. The model achieved a BLEU score of 8.72, indicating a moderate level of translation accuracy. This score reflects the model's ability to generate translations that are reasonably close to human reference translations, though further refinements are necessary to enhance performance. This research demonstrates the potential of neural machine translation in handling less-resourced

language pairs and highlights the importance of data preparation and model evaluation in developing effective translation systems.

Keywords; Machine Translation, OpenNMT, Dholuo, Swahili, Model training

1: Introduction

Machine translation (MT) is one of the most successful applications in natural language processing, as exemplified by its numerous practical applications and the number of contributions on this topic at major machine learning and natural language processing venues[1]. It can be defined as a subfield of computational linguistics that investigates the use of computer software to translate text or speech from one natural language to another. Machine translation can also be defined as a branch of computational linguistics that is defined as an automatic process by a computerized system that convert a piece of text (written or spoken) from one natural language referred to as a source language (SL) to another natural language called the target language (TL) with human intervention or not[2]. It is an area of applied research that draws ideas and techniques from linguistics, computer science, Artificial Intelligence (AI), translation theory and statistics. The goal of machine translation is to develop a system that accurately produces a good translation between human languages[3].

Despite recent advances in translation quality for a handful of language pairs and domains, MT systems still perform poorly on low-resource languages, that is, languages without a lot of training data. In fact, many low-resource languages are not even supported by most popular translation engines. Yet, much of the world's population speak low-resource languages and would benefit from improvements in translation quality on their native languages. As a result, the field has been

increasing focusing towards low-resource languages. At present, there are very few benchmarks on low-resource languages [4]. These often have very low coverage of low-resource languages.

Despite advancements in Neural Machine Translation, there is still a need for more research in various low resourced languages. The Dholuo language, predominantly spoken by the Luo community in Kenya, faces significant challenges in digital representation and accessibility, particularly in automated translation to more widely spoken languages like Swahili. Despite Swahili being a national language in Kenya, existing translation tools do not adequately support Dholuo, creating a communication barrier that hampers socio-economic and educational integration. There is a pressing need for a Neural Machine Translator model that can accurately translate Dholuo to Swahili, ensuring inclusivity and fostering better understanding among speakers of these languages.

The development of a Neural Machine Translation (NMT) model for Dholuo to Swahili translation holds substantial significance for several reasons. First, it promotes linguistic inclusivity by enabling Dholuo speakers to access digital communication and services available in Swahili, thus bridging a significant language gap. Additionally, the study aids in cultural preservation by digitizing and translating Dholuo, supporting efforts to maintain and document the language for future generations. This project also enhances educational opportunities by providing resources that help Dholuo-speaking students learn Swahili, thereby improving their language skills and academic prospects. Economically, the translation model facilitates better communication between Dholuo and Swahili speakers, potentially boosting trade, business, and employment within the region. Finally, this research contributes to the broader field of NMT by addressing the unique challenges of low-resource language translation, paving the way for further technological advancements and applications.

The prime focus of this paper was to develop an open source Neural Machine Translator using OpenNMT-py that will be used to translate from Dholuo to Swahili. OpenNMT is an open source ecosystem for neural machine translation and neural sequence learning[5].

2: Related Works

This section provides an overview of some of the research that have been done to come up with various Neural Machine Translation Models. Some of the Techniques that were applied together with the findings that they found.

This paper[6] enumerates the development of Panlingua-KMI Machine Translation(MT) systems for Hindi to Nepali language pair, designed as part of the Similar Language Translation Task at the WMT 2019 SharedTask. The Panlingua-KMI team conducted a series of experiments to explore both the phrase-based statistical (PBSMT) and neural methods (NMT). Among the 11 MT systems prepared under this task, 6 PBSMT systems were prepared for Nepali-Hindi, 1 PBSMT for Hindi-Nepali, and 2 NMT systems were developed for Nepali to Hindi. The results showed that PBSMT could be an effective method for developing MT systems for closely related languages. Also, it was seen that NMT performed better than SMT on fluency level but the relation between source and target language was erroneous, thereby, resulting in poor BLEU score and higher TER. Furthermore, alterations at the pre-processing stage did not render any improvement in SMT systems, thus, strengthening the importance of lower casing and excluding non-UTF characters from the data sets. It was also observed that datasets with a maximum length of sentences of up to 40 words performed better than those with up to 80 words.

In this research[7],introduced the FLORES-101 evaluation benchmark, consisting of 3001 sentences extracted from English Wikipedia and covering a variety of different topics and domains. These sentences were translated into 101 languages by professional translators through

a carefully controlled process. The resulting dataset enabled a better assessment of model quality on the long tail of low-resource languages, including the evaluation of many-to-many multilingual translation systems, as all translations were fully aligned. Unlike many other datasets, FLORES-101 was professionally translated, including human evaluation during dataset creation. Beyond translation, FLORES-101 can be used to evaluate tasks such as sentence classification, language identification, and domain adaptation.

In their research [8] introduced a universal neural machine translation (NMT) system capable of translating between any language pair. They built a single massively multilingual NMT model that can handle 103 languages trained on over 25 billion examples. Their system demonstrated effective transfer learning ability, significantly improving translation quality of low-resourced languages, while keeping high-resource language translation quality with competitive bilingual baselines. They provided an in-depth analysis of various aspects of model building that are crucial to achieving quality and practicality in universal NMT.

In this work,[9] they introduced the FLORES evaluation datasets for Nepali–English and Sinhala–English, based on sentences translated from Wikipedia. Compared to English, these are languages with very different morphology and syntax, for which little out-of-domain parallel data is available and for which relatively large amounts of monolingual data are freely available. They described their process of collecting and cross-checking the quality of translations and reported baseline performance using several learning settings: fully supervised, weakly supervised, semi-supervised, and fully unsupervised. The experiments demonstrated that current state-of-the-art methods performed rather poorly on this benchmark, posing a challenge to the research community working on low-resource MT.

In their research [10] came up with an NMT between English and five African LRL pairs (Swahili, Amharic, Tigrigna, Oromo, Somali [SATOS]). They collected the available resources on the SATOS languages to evaluate the current state of NMT for LRLs. Their evaluation, comparing a baseline single language pair NMT model against semi-supervised learning, transfer learning, and multilingual modeling, showed significant performance improvements both in the En to LRL and LRL to En directions. In terms of averaged BLEU score, the multilingual approach showed the largest gains, up to +5 points, in six out of ten translation directions.

As illustrated in the above work, there was a need for more research especially in the low resourced languages like Dholuo and Swahili. This will not only help the people who speak these languages but also help the researchers understand the challenges encountered when trying to come up with a Neural Machine Translation in low-resourced languages.

3: Methodology

3.1 Introduction

In this research Kencorpus dataset was used. The dataset is a text and speech corpus for three languages predominantly spoken in Kenya: Swahili, Dholuo and Luhya (three dialects of Lumarachi, Lulogooli and Lubukusu)[11]. The Dholuo to Swahili corpus was chosen to come up with the Neural Machine Translator.

There have been a variety of toolkits that can be utilized when coming up with an NMT. In our case we used openMT-py. OpenNMT is an open-source toolkit started in December 2016 for neural machine translation and neural sequence modeling[12]. It is a toolkit that can be easily used with a recommendable accuracy. OpenNMT has three implementations for industrial and academic purposes. They include; OpenNMT-lua developed with LuaTorch, OpenNMT-tf written

with TensorFlow, and OpenNMT-py developed using PyTorch and suitable for research in translation, summary, morphology and other domains[5].

To use OpenNMT –py one needs to have Python ≥ 3.8 , PyTorch $\geq 2.0 < 2.2$, and also it needs to be installed using the pip command.

3.2 Data Preparation

The Dholuo to Swahili dataset from kencorpus comprises of 99 .txt files. These text files needed to be merged to form one text file that has all the data. When merging, we checked whether there was a corresponding Translation of a given source sentence. If not, the sentences were omitted while displaying a warning to indicate there was a mismatch.

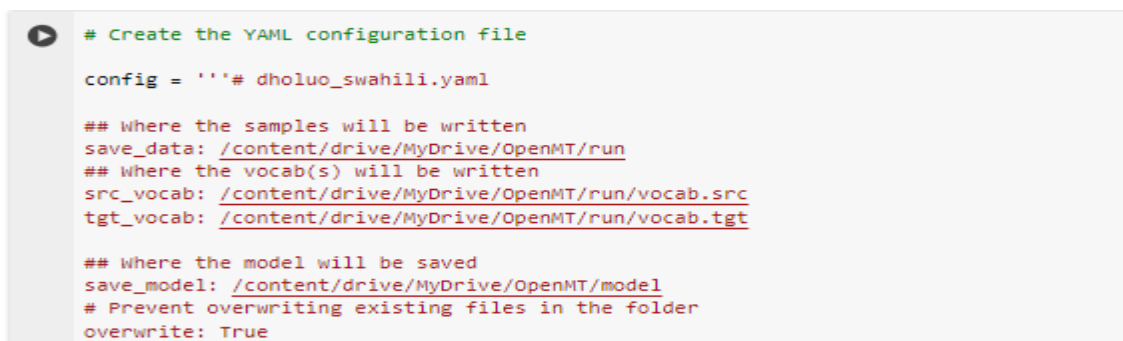
The next part involved confirming that every Original sentence had a translation. After ensuring that each sentence in the source data(Dholuo) had a corresponding translation in Swahili, we went ahead and tried to do some cleaning. This involved using the re (regular expression) library. Some of the things that we did are; Converting the sentences to lowercase, replacing multiple whitespaces with a single space, removing non-alphabetic characters while maintaining commas, spaces, numbers, full stops. Also, ensuring that sentences end with full stop.

The next step involved separating the text file into Original texts(O) and Translated texts(T) and removing the starting O: and T: from each sentence in both files.

3.3 Model Building and Training

To train our model using OpenNMT-py, we needed to split our dataset into Training data, Validation data, and Testing data in both our new files (Files containing the Source sentences and the ones containing the Target sentences). We had 3027 training texts and 929 validation texts. For testing, we had 80 sentences.

To build and run the model, Google Colab was used. Colab is a Google interface that allows you to run Python and bash commands using Jupyter-like notebooks[13]. The main advantage is that everything is installed on their side and GPUs are available. Before training our model, we started by passing our data in a YAML configuration file. This file was used to generate the vocabs that were used by our model. In this file, we specified where our vocabulary will be saved and where the model will be saved highlighted in Figure 1.



```
# Create the YAML configuration file

config = {'# dholuo_swahili.yaml

## Where the samples will be written
save_data: /content/drive/MyDrive/OpenMT/run
## Where the vocab(s) will be written
src_vocab: /content/drive/MyDrive/OpenMT/run/vocab.src
tgt_vocab: /content/drive/MyDrive/OpenMT/run/vocab.tgt

## Where the model will be saved
save_model: /content/drive/MyDrive/OpenMT/model
# Prevent overwriting existing files in the folder
overwrite: True
```

Figure 1: Creating the configuration file

Furthermore, in the config file, we indicate where our datasets are located i.e. the Source files and the target files. Specifying the number of Graphical Processing Unit that will be used its also something that is indicated. Figure 2 is used to indicate how the config file looks.


```
# Corpus opts:
data:
  corpus_1:
    path_src: /content/drive/MyDrive/OpenMT/N_combined_O_train.txt
    path_tgt: /content/drive/MyDrive/OpenMT/N_combined_T_train.txt
  valid:
    path_src: /content/drive/MyDrive/OpenMT/N_combined_O_val.txt
    path_tgt: /content/drive/MyDrive/OpenMT/N_combined_T_val.txt

# Train on a single GPU
world_size: 1
gpu_ranks: [0]
save_checkpoint_steps: 500
train_steps: 1000
valid_steps: 500
early_stopping: 5
...

with open("dholuo_swahili.yaml", "w+") as config_yaml:
    config_yaml.write(config)

!cat dholuo_swahili.yaml
```

Figure 2: Contents of the Configuration file

After giving those configurations, we executed the following code to build the vocabulary.

Building the Vocabulary

```
# Build Vocabulary
!onmt_build_vocab -config dholuo_swahili.yaml -n_sample -1
```

Corpus corpus_1's weight should be given. We default it to 1 for you.
[2024-06-18 07:45:06,564 INFO] Counter vocab from -1 samples.
[2024-06-18 07:45:06,564 INFO] n_sample=-1: Build vocab on full datasets.
[2024-06-18 07:45:07,045 INFO] Counters src: 8835
[2024-06-18 07:45:07,046 INFO] Counters tgt: 9623
[2024-06-18 07:45:07,046 WARNING] path /content/drive/MyDrive/OpenMT/run/vocab.src exists, may overwrite...
[2024-06-18 07:45:07,075 WARNING] path /content/drive/MyDrive/OpenMT/run/vocab.tgt exists, may overwrite...

Figure 3: Building the Vocabulary

To train the model we executed the code shown in Figure 4.

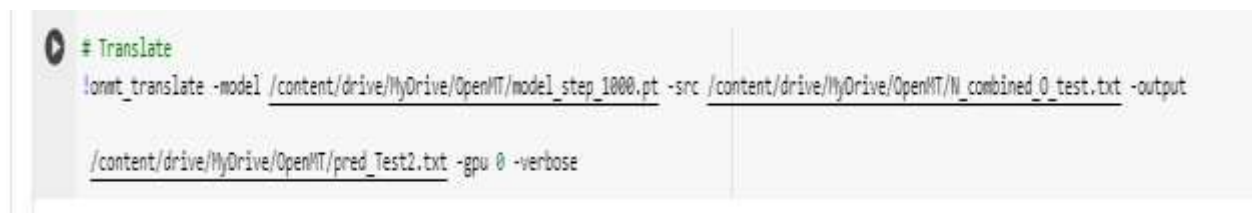
Train the Neural Machine Translator model

```
[ ] # Train the Neural Machine Translator (NMT)
!onmt_train -config dholuo_swahili.yaml
```

Figure 4: Training the NMT

3.4 Model translation

Now having trained the model, we used the test data to assess how well our model performed. In our case, we had 80 sentences for testing. We used the trained model i.e model_step_1000.pt to test our file N_combined_O_test.txt. The output of this process was saved in a file named pred_Test2.txt. Figure 5 illustrates how the test was conducted.



```
# Translate
!onmt_translate -model /content/drive/MyDrive/OpenMT/model_step_1000.pt -src /content/drive/MyDrive/OpenMT/N_combined_O_test.txt -output
/content/drive/MyDrive/OpenMT/pred_Test2.txt -gpu 0 -verbose
```

Figure 5: Translating the Test data

4: Results and Discussion

Figure 6 show some of the translation that were made by the model after training it.

```
SENT 10: ['jamni', 'kod', 'gwen', 'mane', 'ni', 'ei', 'jikon', 'owang', 'duto.']
PRED 10: na marafiki wa kiume na mzee mbaya.
PRED SCORE: -1.6329

[2024-06-18 09:17:21,129 INFO]
SENT 11: ['onge', 'ngat', 'ma', 'owito', 'ngimane', 'kata', 'ma', 'oyudo', 'inyruok', 'emasirano.']
PRED 11: baada ya dakika chache bi mengich kutokana na ugonjwa wa korona.
PRED SCORE: -1.1560

[2024-06-18 09:17:21,129 INFO]
SENT 12: ['ji', '1,004', 'oyud', 'gi', 'corona', 'eseche', '24', 'mokalo', 'epim', 'ma', 'otim', 'ne', 'ji', '6,151.']
PRED 12: watu watatu wameaga kutokana na ugonjwa wa korona katika kaunti ya migori.
PRED SCORE: -0.7325

[2024-06-18 09:17:21,130 INFO]
SENT 13: ['piny', 'owacho', 'ochiwo', 'yuak', 'ewi', 'medruok', 'ekwan', 'masiche', 'mag', 'apaya.']
PRED 13: sheria zilizowekwa kukabiliana na ugonjwa wa korona nchini.
PRED SCORE: -0.6999

[2024-06-18 09:17:21,130 INFO]
SENT 14: ['migosi', 'sonko', 'odonjo', 'ne', 'jangad', 'buche', 'daglas', 'ogoti.']
PRED 14: msheshimiwa mke sonko ametolewa katika stesheni ya maendeleo katika kaunti ya homabay.
PRED SCORE: -0.9140

[2024-06-18 09:17:21,131 INFO]
SENT 15: ['apisa', 'mag', 'thieth', '897', 'oseyudo', 'chanjo', 'mar', 'corona', 'e', 'county', 'ma', 'kisumu.']
PRED 15: maafisa wa afya katika kaunti ya machakos katika kaunti ya migori.
PRED SCORE: -1.2271

[2024-06-18 09:17:21,131 INFO]
SENT 16: ['jolupo', 'enam', 'lolwe', 'ojiw', 'oti', 'kod', 'kiluwa', 'mopuodhi.']
PRED 16: wavuvi 16 wametekwa nyara na wahalifu wasiojulikana kati ya uganda na drc.
PRED SCORE: -0.6365
```

Figure 6: Translation Sample

To evaluate how well the model performed, Predicted score, Predicted Perplexity and Blue Score were used. The predicted score (Pred score) typically refers to the log-likelihood score of the predicted sequences. It is a measure of how likely the predicted sequence (translation) is according to the model. Higher scores indicate that the model is more confident about its predictions. Perplexity is a measure of how well a probabilistic model predicts a sample. It is the exponentiation of the average negative log-likelihood per word. It takes into account the length of the sequences and the probabilities assigned by the model to the predicted tokens[5]. Our model had an average Predicted Score of -1.1367 and Predicted Perplexity of 3.12.

Low perplexity indicates good model performance in predicting the next word in sequences, suggesting the model is relatively confident in its predictions. The less negative predicted score, the better the model's confidence in its translation. Log probabilities are generally negative because probabilities are between 0 and 1, and the logarithm of a number between 0 and 1 is negative. Predicted perplexity of 3.12 is relatively low, suggesting that the model is relatively confident in its predictions. Lower perplexity values generally indicate a better performing model in terms of predicting the next word in a sequence.

Another evaluation metric that we used was Blue Score toolkit which helps to assess the quality of the predictions made with alignment to reference translations. was used[16] in calculating the Blue Score. This tool gave a score of 8.72. This means that our model needs some improvements by either adding of more data or enhancing the cleaning process that we did.

5: Conclusion and Future Work

In this research, we were able to develop a Neural Machine Translator for Dholuo to Swahili. We utilized the kencorpus dataset which required some preprocessing before being utilized.

OpenNMT-py tool was used to train our model which had a Blue score of 8.72. With this performance, we recommend that in future work, more data can be used to train this model or Data Augmentation techniques can be utilized in order to ensure that more data has been used in training.

Also, in the future, we can focus on experimenting with advanced architectures like Transformers and optimizing hyperparameters. Additionally, implementing regularization techniques, and advanced training strategies such as transfer learning are some of the things that will be done.

References

- [1] Q. Wang *et al.*, “Learning Deep Transformer Models for Machine Translation,” Jun. 04, 2019, *arXiv*: arXiv:1906.01787. Accessed: Aug. 05, 2024. [Online]. Available: <http://arxiv.org/abs/1906.01787>
- [2] B. Kituku, L. Muchemi, and W. Nganga, “A Review on Machine Translation Approaches,” *Indones. J. Electr. Eng. Comput. Sci.*, vol. 1, no. 1, p. 182, Jan. 2016, doi: 10.11591/ijeecs.v1.i1.pp182-190.
- [3] J. Oladosu, A. Esan, I. Adeyanju, B. Adegoke, O. Olaniyan, and B. Omodunbi, “Approaches to Machine Translation: A Review,” *FUOYE J. Eng. Technol.*, vol. 1, no. 1, Sep. 2016, doi: 10.46792/fuoyej.v1i1.26.
- [4] W. Nekoto *et al.*, “Participatory Research for Low-resourced Machine Translation: A Case Study in African Languages,” Nov. 06, 2020, *arXiv*: arXiv:2010.02353. Accessed: Jun. 18, 2024. [Online]. Available: <http://arxiv.org/abs/2010.02353>
- [5] G. Klein, Y. Kim, Y. Deng, V. Nguyen, J. Senellart, and A. M. Rush, “OpenNMT: Neural Machine Translation Toolkit,” May 28, 2018, *arXiv*: arXiv:1805.11462. Accessed: Jun. 18, 2024. [Online]. Available: <http://arxiv.org/abs/1805.11462>
- [6] A. Kr. Ojha, R. Kumar, A. Bansal, and P. Rani, “Panlingua-KMI MT System for Similar Language Translation Task at WMT 2019,” in *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*, Florence, Italy: Association for Computational Linguistics, 2019, pp. 213–218. doi: 10.18653/v1/W19-5429.
- [7] N. Goyal *et al.*, “The FLORES-101 Evaluation Benchmark for Low-Resource and Multilingual Machine Translation,” *Trans. Assoc. Comput. Linguist.*, vol. 10, pp. 522–538, May 2022, doi: 10.1162/tacl_a_00474.
- [8] N. Arivazhagan *et al.*, “Massively Multilingual Neural Machine Translation in the Wild: Findings and Challenges,” Jul. 11, 2019, *arXiv*: arXiv:1907.05019. Accessed: Jun. 18, 2024. [Online]. Available: <http://arxiv.org/abs/1907.05019>

- [9] F. Guzmán *et al.*, “The FLoRes Evaluation Datasets for Low-Resource Machine Translation: Nepali-English and Sinhala-English,” Sep. 14, 2019, *arXiv*: arXiv:1902.01382. Accessed: Jun. 18, 2024. [Online]. Available: <http://arxiv.org/abs/1902.01382>
- [10] S. M. Lakew, M. Negri, and M. Turchi, “Low Resource Neural Machine Translation: A Benchmark for Five African Languages,” Mar. 31, 2020, *arXiv*: arXiv:2003.14402. Accessed: Jun. 18, 2024. [Online]. Available: <http://arxiv.org/abs/2003.14402>
- [11] B. Wanjawa, L. Wanzare, F. Indede, O. McOnyango, E. Ombui, and L. Muchemi, “Kencorpus: A Kenyan Language Corpus of Swahili, Dholuo and Luhya for Natural Language Processing Tasks,” *J. Lang. Technol. Comput. Linguist.*, vol. 36, no. 2, pp. 1–27, Jun. 2023, doi: 10.21248/jlcl.36.2023.243.
- [12] G. Klein, Y. Kim, Y. Deng, V. Nguyen, J. Senellart, and A. M. Rush, “OpenNMT: Neural Machine Translation Toolkit,” vol. 1, 2018.
- [13] C. Vidal-Silva *et al.*, “Developing Computing Competencies Without Restrictions,” *IEEE Access*, vol. 10, pp. 106568–106580, 2022, doi: 10.1109/ACCESS.2022.3211973.
- [14] “Investigation Role of AI in MT final thesis.pdf.”
- [15] C. Vidal-Silva *et al.*, “Developing Computing Competencies Without Restrictions,” *IEEE Access*, vol. 10, pp. 106568–106580, 2022, doi: 10.1109/ACCESS.2022.3211973.
- [16] letsmt- BLEU. (2024). <https://www.letsmt.eu/Bleu.aspx>