Baseline Transliteration Corpus for Improved English-Amharic Machine Translation

Yohannes Biadgligne[1], Kamel Smaili[2]
[1]Sudan University of Science and Technology (SUST) and Bahir Dar Institute of Technology (BIT), Khartoum and Bahir Dar
[2]Loria - Université Lorraine, France
E-mail: yohannesb2001@gmail.com, kamel.smaili@loria.fr

Keywords: machine translation, amharic language, transliteration, augmentation, NMT, RNN, GRU, transformer

Received: September 19, 2022

Machine translation (MT) between English and Amharic is one of the least studied and, performance-wise, least successful topics in the MT field. We therefore propose to apply corpus transliteration and augmentation techniques in this study to address this issue and improve MT performance for the language pairs. This paper presents the creation, the augmentation, and the use of an Amharic to English transliteration corpus for NMT experiments. The created corpus has a total of 450,608 parallel sentences before preprocessing and is used to train three different NMT architectures after preprocessing. These models are actually built using Recurrent Neural Networks with attention mechanism (RNN), Gated Recurrent Units (GRUs), and Transformers. Specifically, for Transformer-based experiments, three different Transformer models with different hyperparameters are created. Compared to previous works, the BLEU score results of all NMT models used in this study are improved. One of the three Transformer models, in particular, achieves the highest BLEU score ever recorded for the language pairs.

Povzetek: Raziskava se ukvarja z izboljšanjem strojnega prevajanja (MT) med angleščino in amharščino, eno izmed najmanj proučevanih in uspešnih področij v MT. Predlagana je uporaba tehnike transliteracije in razširjanja korpusov. Izdelali so korpus za preizkušanje NMT, ki obsega 450,608 paralelnih stavkov.

1 Introduction

In today's modern age of technology and social media, it is increasingly common to incorporate foreign words into one's native tongue and compose in one language using scripts from other languages. English is the most widely used language in this regard [1]. This can be attributed to many reasons, but one of them is the prevalence of the 'QWERTY' keyboard layout in laptops, smartphones, and even mechanical typewriters, especially in developing countries. Thus, many people who don't speak English prefer to compose their ideas using English scripts across multiple messaging platforms. This writing method is known as transliteration [2, 3].

In the 1990s, NLP researchers were interested in creating machines for transliteration purpose to support other research areas. This was the first time the concept of machine transliteration was introduced. Machine transliteration is a subfield of MT and cross-language information retrieval (CLIR). Its primary goal is to use computers to convert a text from one language script (the source language) to another language script (the target language) while maintaining as much pronunciation as possible. In technical terms, it is concerned with accurately representing the graphemes of one language script using the script of another language [4].

The literature on MT suggests that transliteration can be used with MT systems to reduce translation errors and improve precision when translating names (named entities), technical terms, and loan (borrowed) words [5, 6, 7]. Particularly for languages with limited resources (e.g small bilingual corpora), such as Amharic. Because learning all of the words of a given language from a small amount of bilingual training data is impossible [8, 9, 10]. Finch et al.[11], carried out a large-scale real-world evaluation of the use of automatic transliteration in an MT system and demonstrated that using a transliteration system can improve MT quality when translating unknown words. As a result, machine transliteration has become a promising application for the use of MT. Table 1 shows the distinction between translation and transliteration for the languages under consideration (Amharic and English).

Table 1: Example of Amharic to English translation and transliteration

		Transliteration
Amharic	**Translation**	
ኢትዮጵያ	Ethiopia	ītiyop'iya
አፍሪካ	Africa	āfirīka
ውስጥ ናት	is in	wisit'i nati

Amharic (አማርኛ/əmərɨgnə), the main language of Ethiopia, has its own scripts and is the second most widely spoken Semitic language after Arabic. The Amharic script was originally derived from Ge'ez (ግእዝ/gə'əzzə). Although it has disappeared as a colloquial language, Gee'z is the main language used for prayer, ritual performance, and the main teaching language in the Ethiopian Orthodox Church [12]. Amharic uses a slightly modified version of the Gee'z alphabet. It consists of 34 basic characters, each of which has seven forms depending on which vowels in syllables are pronounced. Even though it is no longer widely used, Amharic also inherits all the Gee'z numeric character sets [13].

2 Related work

Machine transliteration is rarely an end goal by itself, but is often used as part of other NLP tasks (such as CLIR, QA, or MT). In light of its importance in these fields, a number of transliteration mechanisms have been proposed for non-English languages including Russian, Chinese, Korean, Arabic, Persian, and Indian [14]. These mechanisms generally fall into three broad categories: linguistic (rule-based) approaches; statistical approaches; and deep learning approaches [15].

The linguistic approach uses hand-crafted rules based on pattern matching, which needs a linguistic analysis to formulate rules. This approach requires a thorough understanding of the language under consideration. Early attempts used this method to construct baseline transliteration corpora, and it is still used as a starting point to acquire transliteration corpora for low-resource languages [16].

Deep and Goyal [17] have proposed a Punjabi to English transliteration system that uses a linguistic-based approach. In the proposed transliteration scheme, a grapheme-based method is used to model the transliteration problem and achieves an accuracy of 93.22% when transliterating common names. A similar transliteration system has been developed by Goyal and Lehal [18] by implementing fifty complex rules. Their system was found to give about 98% accuracy for transliterating proper names, city names, country names, subject-related technical terms etc.

Various transliteration systems were proposed during the Named Entities Workshop (NEWs) evaluation campaigns between 2009 and 2018 [19]. During the campaigns, transliteration is done from English into various languages with various writing systems. As a result of this workshop, many advances have been made in methodologies for transliterating proper nouns. There have been several approaches developed, including grapheme-to-phoneme conversion [20, 21], based on statistics like machine translation [16, 22], as well as neural networks, such as sequence-to-sequence models and Long-Short-Term-Memory (LSTM) [23, 24, 25, 26, 27].

The three transliteration approaches discussed previously can be based on grapheme [1], phoneme [2], hybrid, or correspondence transliteration models.

- **Grapheme-based models:** directly converts source language graphemes into target language graphemes without requiring phonetic knowledge of the source language words.
- **Phoneme-based models:** uses source language phonemes as a pivot when producing target language graphemes from source language graphemes.
- **Hybrid and Correspondence-based models:** use both source language graphemes and phonemes.

Generally, statistical and neural network techniques based on large parallel transliteration corpora work well for rich-resource languages but low-resource languages do not have the luxury of such resources. For such languages, rule-based transliteration is the only viable option [16].

2.1 Amharic transliteration

In our literature review, we found two cases where Amharic was studied for transliteration tasks. The first attempt was made by Tadele Tedla [28]. His objective was to develop a framework to convert ASCII transliterated Amharic text to the original Amharic text. In the transliteration of three random test data-sets, the model achieves respectively 97.7, 99.7, and 98.4 percent accuracy. The first set of test data consists of an ASCII transliterated Amharic word list of 32,482 words. The second set of test data is a transliterated poem with 1277 words, and the third set of data is a recipe for Injera, a common local food in Ethiopia, with 123 transliterated Amharic words.

Gezmu et al. [29] is the second attempt at Amharic-to-English machine transliteration. In their work, they used machine transliteration as a tool (to facilitate vocabulary sharing) to improve the performance of Amharic-English MT. Despite claiming to have created an Amharic-English transliteration corpus for named entities and borrowed words, they did not make it publicly available. Based on a review of the literature, we believe that our attempt is the first to create a large Amharic-English transliteration corpus for the English-Amharic NMT.

3 Motivation

Developing a reliable English-Amharic MT system remains a challenge. A scarcity of resources and the absence of well-organized MT research projects are the two major obstacles to overcoming this challenge. Our search reveals that the majority, if not all, of the research works

[1] A grapheme is a letter or set of letters that represent the sound (phoneme) of a word.

[2] One of the smallest speech units that distinguishes one word from another is a word.

on English-Amharic MT are done by independent individuals and are disjointed. The BLEU score results for these language pairs are, therefore, not indicative of high quality translation, according to a general interpretation of the BLEU score. Thus, this study aims to enhance English-Amharic MT performance by incorporating transliteration as a tool. To achieve this goal, we created an Amharic-English transliteration corpus from previously collected English-Amharic MT corpus [30, 31] and used it for English-Amharic NMT experiments. This is the first baseline corpus for these language pairs, which will be made available to MT and IR researchers.

4 Experimental set-up

4.1 Corpus preparation

The objective of this study is to improve the performance of English-Amharic MT by using a transliterated and augmented corpus. However, the data required for training the NMT models is not available. As a result, the previously gathered English-Amharic translation corpus is used to generate an Amharic-English transliteration corpus. Therefore, this section is devoted to explaining the methods and techniques used to create this corpus, as well as the NMT experiments performed with it.

4.1.1 Acquisition of the previously collected translation corpus

The freely available English-Amharic translation corpus was obtained from the Github Repository [3] [4] [30]. This corpus was compiled from religious, legal, and news domains and contains 225,304 English-Amharic parallel sentences.

4.1.2 Pre-transliteration preprocessing

This step is completed before the transliteration process begins. It is performed on the previously acquired original Amharic translation corpus. Normalization of homophone characters, removal of punctuation marks, and conversion of Amharic to Arabic numerals are all carried out. After these preprocessing tasks are completed, the corpus is divided into 25 parts and distributed to data collectors. These data collectors use Google Translate to transliterate Amharic sentences into English scripts, and then they collect these sentences by copying and pasting them into a text file.

4.1.3 Transliterating the acquired corpus

For the successful completion of this task two different steps are followed.

1. **Performing transliteration:** This process was carried out using Google's online translation tool [5]. Regardless of its primary goal of translation, Google Translate can generate text transliterations as part of the translation process if the two languages use distinct scripts. The main task completed at this stage, as shown in Figure 1, was transliterating Amharic sentences to English using Google Translate and collecting the transliterated sentences.

   In order to transliterate and compile a total of 225,304 Amharic sentences, 25 data collectors (computer science students) participated. The entire process of transliterating and normalizing these Amharic sentences takes 60 days, and each data collector has a daily throughput of 150 sentences. Prior to the transliteration task, each data collector was provided with brief training and guidance to improve the quality and consistency of the transliteration process.

2. **Normalizing the transliteration corpus:** After the transliteration corpus was collected, the next task accomplished was corpus normalization. The objective of this task was to make the transliterations of Amharic loan words and named entities (NEs) as close as possible to the spelling of English words, so that they become useful for MT purposes. To assist this manual normalization process, true casing is carried out first using Moses' built-in true-caser script. Because Amharic has a Subject-Verb-Object (SVO) grammatical structure and NEs are more likely to appear at the beginning of a sentence, true casing allowed us to capitalize the first letter of the majority of NEs. This reduces the amount of work required to locate and correct NEs when they are transliterated differently than the English version. Table 3 contains examples of transliterations produced by Google Translate and their normalized forms. The table also includes the Levenshtein edit distance [32] computed between the English translation and the generated and normalized transliterations. Computing the Levenshtein edit distance allows us to choose the transliterations closest to the English translation.

   As depicted in the table, all of the differences between the English translation and the generated transliterations (using Google Translate) occur in representing the sixth form of Amharic characters. For instance, the name **Daniel** (ዳንኤል) is spelled as **Dani'ēli** by Google Translate. But its correct English spelling (English translation) is **Daniel**. This discrepancy occurs at writing the sixth form of Amharic characters. In the above example Google Translate uses **ni** and **li** to represent (ን) and (ል) respectively. So, to make the transliterated loan words and named entities in the corpus closer to the English word these characters are normalized to (**n**) (ን) and (**l**) (ል). This normalization is done for all

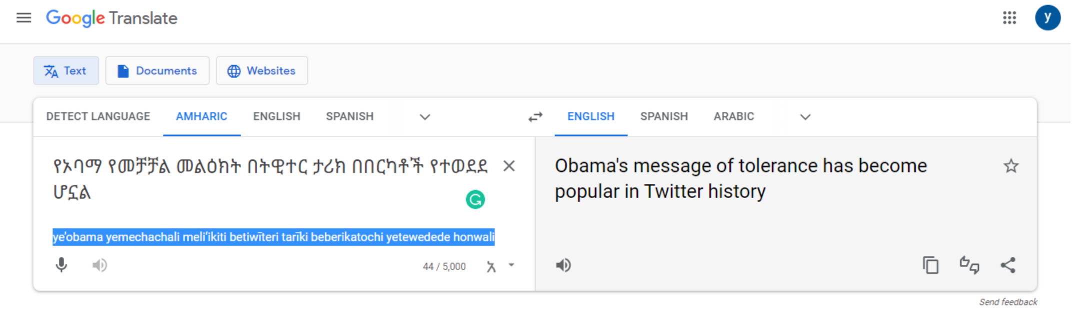
[3] https://github.com/yohannesb/English-Amharic-origional-corpus

[4] https://github.com/yohannesb/English-Amharic-Augmented-corpus

[5] https://translate.google.com/

Table 2: Summary of related works

Author, Year	Language pairs	Model, Approach used	Objective	Results
Goyal, Vishal, and Gurpreet Singh Lehal, (2009)	Hindi to Punjabi	G2P, Rule-based approach	For MT	98 % accuracy.
Deep, Kamal, and Vishal Goyal, (2011)	Punjabi to English	G2P, Rule-based approach	For MT and CLIR	93.22 % accuracy.
Laurent, Antoine, Paul Deleglise, and Sylvain Meignier, (2009)	French to French	G2P, SMT approach	For ASR system	Comparable to the dictionary look-up strategy.
Finch, Andrew, and Eiichiro Sumita, (2010)	English to (Thai, Hindi, Tamil, kannada, Japanese, Bangla)	G2), PBSMT and Joint Multigram model	For MT and CLIR	Better performance from other previous works.
Yao, Kaisheng, and Geoffrey Zweig, (2015)	US English	G2P, Bi-LSTM	For MT and Image captioning	Outperforms previous SOTA models.
Rao, Kanishka, et al. , (2015)	US English	G2P, LSTM-RNN	Not explicitly mentioned	Improvement over previous similar works.
Shao, Yan, and Joakim Nivre, (2016)	English to Chinese and Chinese to English	Neural Networks (CNN and RNN),	Not explicitly mentioned	Achieved competitive results with SOTA models at the time.
Thu, Ye Kyaw, et al., (2016)	Maynamar to English	G2P, PBSMT , CRF, S-Arrow, JSM	For pronunciation dictionary	CRF and PBSMT achieved best results.
Tedla, Tadele, (2015)	ASCII transliterated Amharic to original Amharic letter	G2P, key map dictionary	Not explicitly mentioned	Ranges from 97.7% to 99.7% accuracy.
Gezmu, Andargachew Mekonnen, Andreas Nurnberger, and Tesfaye bayu Bati, (2021)	Amharic to English	G2P, Rule-based approach	For MT	Better results than previous works.



የኦባማ የመቻቻል መልዕክት በትዊተር ታሪክ በበርካቶች የተወደደ ሆኗል

ye'obama yemechachali meli'ikiti betiwīteri tarīki beberikatochi yetewedede honwali

Obama's message of tolerance has become popular in Twitter history

Figure 1: Snapshot taken from Google Translate.

sixth form characters of Amharic. The transliteration character map used in this work is depicted by Table 4. Which is the modified version of the United Nations Romanization Systems for Geographical Names (BGN/PCGN 1967 System) approved for Amharic to English transliteration [33]. Actually, in this standard the six form of Amharic characters have two optional representations.

Overall according to the Levenshtein edit distance, the normalized form of Google transliteration is closer to the English translation.

4.1.4 Post-transliteration preprocessing

At this stage, cleaning and splitting of the corpus are performed. These two preprocessing techniques make the transliterated corpus ready for MT training purposes. The cleaning task removes empty lines from the corpus, avoids redundant space between characters and words, and cuts and discards extremely long sentences (sentences with more than 80 words). As a result, after completing this task, the total number of sentences in the corpus drops from 225,304 to 218,365.

Finally, for training our MT models, the transliterated and preprocessed texts are divided into three parts. For the

Table 3: Example of Amharic to English transliteration using Google translate and normalized form of the transliteration.

Amharic text	English translation	Google transliteration	Normalized form of Google transliteration	Levenshtein edit-distance b/n English and Google)	Levenshtein edit-distance b/n English and normalized form	Levenshtein edit-distance (English and Amharic
ዳንኤል	daniel	dani’ēli	danēl	3	2	6
ሞሀመድ	mohamed	moḥāmedi	mohāmed	2	0	7
አይሻ	ayisha	āyisha	āysha	1	1	6
ማርታ	marta	marita	marta	1	0	5
ቤተልሄም	bethlehem	betelihemi	betelhem	3	2	9
ኢትዮጵያ	ethiopia	ītiyop’iya	ityop’ya	4	4	8
ኮምፒዩተር	computer	komipīyuteri	kompīyuter	5	3	8

sake of comparison (to see the effects of transliterated data on the performance of MT models), the same split ration as the experiments done in [31] is used. There are 212,115 sentences for training, 5000 sentences for validation, and 1250 sentences for testing.

4.2 Augmentation of transliterated corpus

In addition to the transliteration task, corpus augmentation is performed to increase the size of the transliterated English-Amharic corpus. Several publications have indicated that corpus augmentation can be an effective method of scaling up corpora, especially for languages with a limited resource base. Hence, in this work, token-level corpus augmentation is applied and the augmented corpus is used as the training dataset for different NMT models. Among alternative token level augmentation techniques random insertion, replacement, deletion, and swapping approaches are selected and implemented. In doing so, seven different augmented corpora are generated by varying the values of (delete probability, replacement probability, and swapping range). Then Cosine similarity between the original corpus and the augmented ones is calculated, and the augmented corpus that preserves approximately 90% of the meaning is selected [31].

The augmentation task is done for training, validation, and test sets to avoid overlapping sentences in each set. By combining these augmented data sets with the transliterated corpus, 424,230 training, 10,000 validation, and 2500 testing sets are created. Overall, this resulted in 436,730 cleaned, transliterated, and augmented sentences.

4.3 NMT Experiments

In this experiment, three different NMT models are created and their performance are evaluated by comparing them to previous attempts for the language pairs. RNN with attention mechanisms, GRU-based, and Transformer-based NMT models were developed, and each model was trained using a transliterated and augmented corpus.

4.3.1 Attention based RNN model:

An open source toolkit called Open-NMT [34] [35] is used to build this model. Given the corpus is divided into three parts (training, validation and testing sets) in the preporcessing stage of this experiment, the first task in training the RNN based model is performing Byte Pair Encoding (BPE). BPE enables NMT model translation on open-vocabulary by encoding rare and unknown words as sequences of sub-word units. This is based on an intuition that various word classes are translatable via smaller units than words [36]. The next step is preprocessing; actually it computes the vocabularies given the most frequent tokens, filters too long sentences, and assigns an index to each token. Finally, RNN based NMT model with attention mechanisms is trained with the parameters depicted in Table 5. Actually, training is the most time consuming task in the whole process of creating this model. A larger batch size is advantageous for improving training time and quality. As a result, a large batch size is used in this experiment. The larger the batch size, the greater the efficiency (matrix multiplication with small batch sizes is very inefficient). Because a larger matrix can more effectively utilize GPU cores and RAM [37] [38].

4.3.2 GRU based model:

In comparison to conventional RNN and LSTM, GRUs are relatively new architectures that are being used in many machine learning applications. Due to their fewer parameters, they improve the training time of LSTM and resolve vanishing and exploding gradients, which occur with RNNs [39].

In order to conduct the GRU-based NMT experiment, three distinct units (encoder, attention, and decoder) are cre-

Table 4: Amharic to English transliteration character map.

	1^{st} Form	2^{nd} Form	3^{rd} Form	4^{th} Form	5^{th} Form	6^{th} Form	7^{th} Form
1	ሀ hā	ሁ hu	ሂ hī	ሃ ha	ሄ hē	ህ hi	ሆ ho
2	ለ le	ሉ lu	ሊ lī	ላ la	ሌ lē	ል li	ሎ lo
3	ሐ hā	ሑ hu	ሒ hī	ሓ ha	ሔ hē	ሕ hi	ሖ ho
4	መ me	ሙ mu	ሚ mī	ማ ma	ሜ mē	ም mi	ሞ mo
5	ሠ še	ሡ šu	ሢ šī	ሣ ša	ሤ šē	ሥ ši	ሦ šo
6	ረ re	ሩ ru	ሪ rī	ራ ra	ሬ rē	ር ri	ሮ ro
7	ሰ se	ሱ su	ሲ sī	ሳ sa	ሴ sē	ስ si	ሶ so
8	ሸ she	ሹ shu	ሺ shī	ሻ sha	ሼ shē	ሽ shi	ሾ sho
9	ቀ k’e	ቁ k’u	ቂ k’ī	ቃ k’a	ቄ k’ē	ቅ k’i	ቆ k’o
10	በ be	ቡ bu	ቢ bī	ባ ba	ቤ bē	ብ bi	ቦ bo
11	ቨ ve	ቩ vu	ቪ vī	ቫ va	ቬ vē	ቭ vi	ቮ vo
12	ተ te	ቱ tu	ቲ tī	ታ ta	ቴ tē	ት ti	ቶ to
13	ቸ che	ቹ chu	ቺ chī	ቻ cha	ቼ chē	ች chi	ቾ cho
14	ኀ hā□	ኁ hu□	ኂ hī□	ኃ ha□	ኄ hē□	ኅ hi□	ኆ ho□
15	ነ ne	ኑ nu	ኒ nī	ና na	ኔ nē	ን ni	ኖ no
16	ኘ nye	ኙ nyu	ኚ nyī	ኛ nya	ኜ nyē	ኝ nyi	ኞ nyo
17	አ ‘ā	ኡ ‘u	ኢ ‘ī	ኣ ‘a	ኤ ‘ē	እ ‘i	ኦ ‘o
18	ከ ke	ኩ ku	ኪ kī	ካ ka	ኬ kē	ክ ki	ኮ ko
19	ኸ he□	ኹ hu□	ኺ hī□	ኻ ha□	ኼ hē□	ኽ hi□	ኾ ho□
20	ወ we	ዉ wu	ዊ wī	ዋ wa	ዌ wē	ው wi	ዎ wo
21	ዐ ‘ā	ዑ ‘u	ዒ ‘ī	ዓ ‘a	ዔ ‘ē	ዕ ‘i	ዖ ‘o
22	ዘ ze	ዙ zu	ዚ zī	ዛ za	ዜ zē	ዝ zi	ዞ zo
23	ዠ zhe	ዡ zhu	ዢ zhī	ዣ zha	ዤ zhē	ዥ zhi	ዦ zho
24	የ ye	ዩ yu	ዪ yī	ያ ya	ዬ yē	ይ yi	ዮ yo
25	ደ de	ዱ du	ዲ dī	ዳ da	ዴ dē	ድ di	ዶ do
26	ጀ je	ጁ ju	ጂ jī	ጃ ja	ጄ jē	ጅ ji	ጆ jo
27	ገ ge	ጉ gu	ጊ gī	ጋ ga	ጌ gē	ግ gi	ጎ go
28	ጠ t’e	ጡ t’u	ጢ t’ī	ጣ t’a	ጤ t’ē	ጥ t’i	ጦ t’o
29	ጨ ch’e	ጩ ch’u	ጪ ch’ī	ጫ ch’a	ጬ ch’ē	ጭ ch’i	ጮ ch’o
30	ጰ p’e	ጱ p’u	ጲ p’ī	ጳ p’a	ጴ p’ē	ጵ p’i	ጶ p’o
31	ጸ ts’e	ጹ ts’u	ጺ ts’ī	ጻ ts’a	ጼ ts’ē	ጽ ts’i	ጾ ts’o
32	ፀ ts□’ e	ፁ ts□’u	ፂ ts□’ī	ፃ ts□’a	ፄ ts□’ē	ፅ ts□’i	ፆ ts□’o
33	ፈ fe	ፉ fu	ፊ fī	ፋ fa	ፌ fē	ፍ fi	ፎ fo
34	ፐ pe	ፑ pu	ፒ pī	ፓ pa	ፔ pē	ፕ pi	ፖ po

Table 5: Parameters and values of RNN model

Parameters	Values
Training set	424,230
Validation set	10000
Testing set	2500
Hidden units	512
Layers	6
Word vec size	512
Train steps	20000
Batch size	4096
Label smoothing	0.1
Attention mechanism	Bahdanau
Evaluation Metric	BLEU

ated. Each of the encoder and decoder units has three GRU layers, with a hidden state size of 512. Before the training begins the tokenizer converts each word to a unique integer value, which is then converted to word embeddings by the embedding unit. The embedding layer has a dimension of 128. The entire architecture of our GRU based NMT model and the detailed training parameters are depicted in Figure 2 and Table 6 respectively.

Table 6: Parameters and values of GRU model

Parameters	Values
Training Set	424,230
Validation Set	10000
Testing Set	2,500
Encoder Units	512
Attention mechanism	Bahdanau
Decoder Units	512
Embedding size	128
Loss function	cross entropy
Optimizer	RMSprop
Batch Size	512
Evaluation Metric	BLEU

4.3.3 Transformer based model

Transformer is architecturally distinct from other NMT models. Because, it is entirely dependent on the attention mechanisms. This makes it suitable for capturing the long-term dependency between words in a given text. In this experiment, Transformer-based models using the NMT-Keras toolkit is built. It is a versatile toolkit based on Keras library for training deep learning NMT models [40]. For comparison purposes three different Transformer models are created : Transformer-Big, Transformer-Default, and Transformer-Best Practice. Hereafter, they are referred as Transformer-B, Transformer-D, and Transformer-BP, respectively.

These models are trained using different hyper-parameter values but the same training, validation, and testing data sets. Transformer-B and Transformers-D are trained using pre-configured hyper-parameter values, whereas Transformer-BP is trained using tuned hyper-parameter values.

In order to determine the hyper-parameter values for Transformer-BP model, several papers that investigates the effect of hyper-parameter values on the translation quality of NMT models are surveyed. More importantly, the papers focuses on Transformer-based models for low-resource language translation are critically reviewed. By considering the size of the corpus the hyper-parameter values are determined. The parameter values of the three models are summarized in Table 7.

5 Experimental results

Table 8 presents all the BLEU score results for the three models. The BLEU score results indicated in augmented corpus column are cited from previous works for the purpose of comparison and analysis.

As shown in the table, the BLEU score results of all the three models are improved due to the utilization of transliterated and augmented corpus. Especially, Transformer-BP and GRU models benefited slightly more from the transliteration corpus than the other models. This is due to the fact that Transformer-BP is trained with hyperparameters that have been adjusted to account for the size of the corpus. While GRU is inherently uses small number of parameters to train, making it easier to select more appropriate hyper-parameter values and achieve better BLEU score results.

On the other hand, the hyper-parameter values for other Transformer based models (Transformer-B and Transformer-D) are set for bigger corpus sizes. So, their performance is lower than all the remaining models. This makes them the least benefited models of the Transliterated corpus.

In general, a T-test (two-tailed) is used to determine whether or not the BLEU score results obtained by models trained with the transliterated corpus are statistically significant. According to the calculation, the t-value (0.000301279) is smaller than the critical value P (0.05), thus indicating there is a significant difference between the two BLEU score values. From this, we can conclude that transliterating the corpus improves the performance of all three NMT models. Especially, the BLEU sore result of the Transformer-BP model is the highest score so far for English-Amharic MT.

6 Conclusion

Low resource MT is still a work in progress and in its crawling stage for a variety of reasons. On the contrary, MT research for resource-rich languages goes a long way in the acquisition of resources and the creation of different MT architectures. As a result, different successful NMT architectures are introduced. These includes RNNs, GRUs

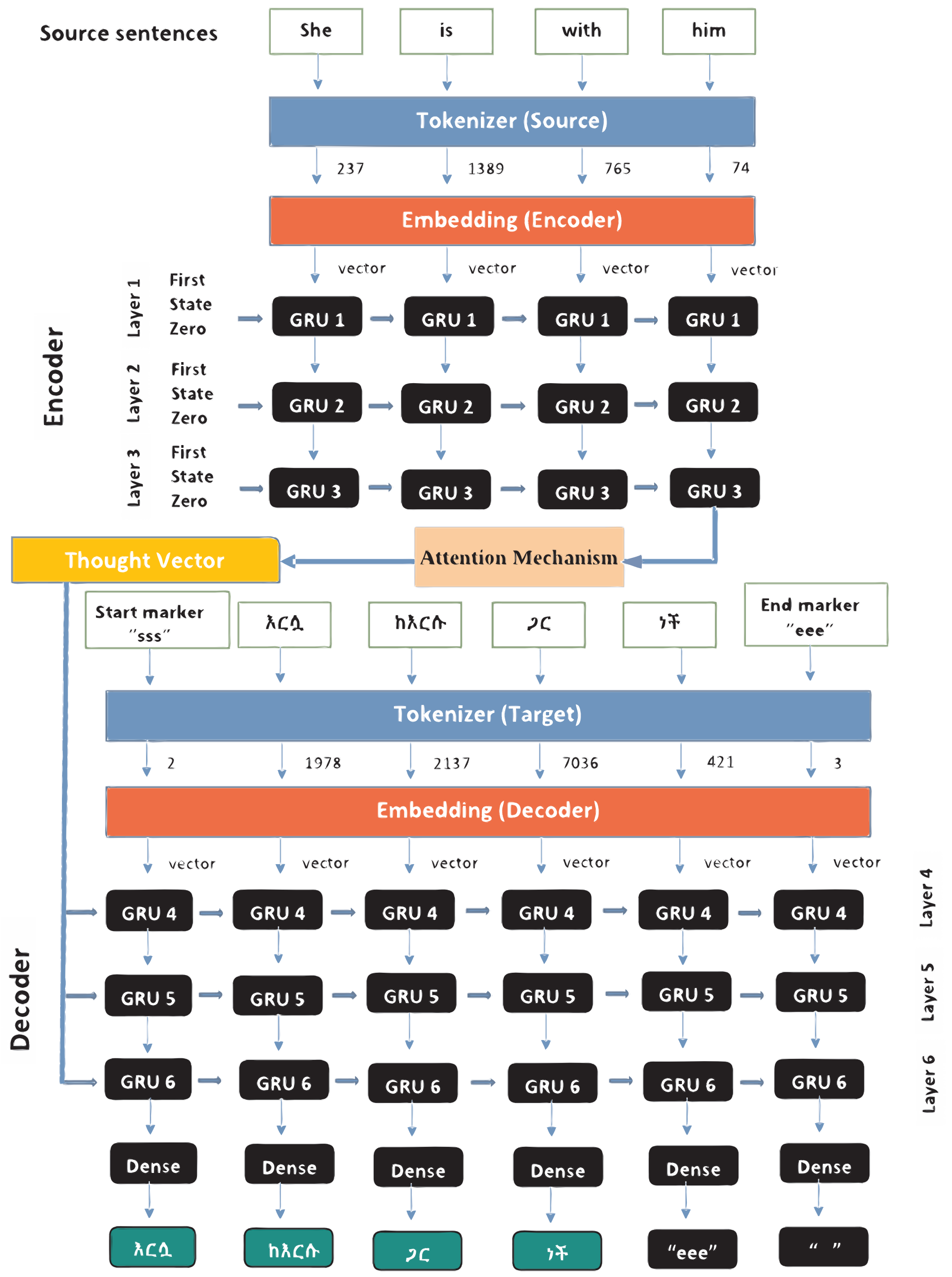


Figure 2: GRU model architecture.

Table 7: Hyperparameters of our Transformer model

Hyper-parameters	**Transformer-B**	**Transformer-D**	**Transformer-BP**
Training set	424,230	424,230	424,230
Validation set	10000	10000	10000
Testing set	2500	2500	2500
feed-forward dimension	4096	2048	2048
BPE size	40k	37k	30k
attention heads	16	8	4
dropout	0.5	0.1	0.3
layers	7	6	5
label smoothing	0.8	0.1	0.3
enc/dec layerDrop	0.4	0.0/0.0	0/0.1
src/tgt word dropout	0.3	0.0/0.0	0.2/0.2
activation dropout	0.5	0.0	0
batch size	12288	4096	12288

Table 8: Experimental results of the different NMT models

Model type	Corpus used	
	Augmented corpus (previous works)	Augmented + Transliterated corpus (present work)
RNN .att	35.38	35.76
GRU	37.79	38.22
Transformer-B	35.62	35.91
Transformer-D	36.53	36.85
Transformer-BP	39.21	39.67

and most importantly Transformer. However, due to resource constraints (particularly lack of huge bilingual corpora), most languages in the low resource language category, are not benefiting from these successful architectures. Amharic is one of these languages. So, in this work, we decided to take up this challenge and attempted to improve the performance of English-Amharic MT using corpus transliteration and augmentation.

For that, we created the biggest Amharic - English transliteration corpus from the previously collected English-Amharic parallel corpus using Google Translate (for transliteration) and human data collectors (for normalization). In the normalization process, transliterated names and borrowed words are spelled as closely to their English translation as possible. After this, token level corpus augmentation technique is applied on the transliterated corpus in order to artificially increase the size of the corpus. By doing so we are able to create a corpus (transliterated and augmented) with a size of 450,608 parallel sentences.

With the created data set RNN with attention mechanism, GRU-based and Transformer based NMT architectures are trained. Compared to a previous work in which we used a corpus augmentation with similar training parameters, all three models in this study achieve better MT performance. Especially, the BLEU score achieved by one of the three models (Transformer-BP) is the state of the art result (39.67 BLEU) for the language pairs so far as much as our knowledge is concerned. Transliteration played a part in this.

Generally, this work adds two contributions to the knowledge base of English-Amharic MT research. The first one is the creation of English-Amharic transliteration and augmentation corpus. The second one is the improvement of English-Amharic MT performance.

References

[1] Kirkpatrick, Andy. English as a lingua franca in ASEAN: A multilingual model. Vol. 1. Hong Kong University Press, 2010. https://doi.org/10.5790/hongkong/9789888028795.003.0008

[2] Kramsch, Claire. ”Teaching foreign languages in an era of globalization: Introduction.” The modern language journal 98.1 (2014): 296-311. https://doi.org/10.1111/j.1540-4781.2014.12057.x

[3] Coulmas, Florian. Sociolinguistics: The study of speakers’ choices. Cambridge University Press, 2013.

[4] Kumaran, Adimugan, and Tobias Kellner. ”A generic framework for machine transliteration.” Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval. 2007. https://doi.org/10.1145/1277741.1277876

[5] Zhou, Dong, et al. ”Translation techniques in cross-language information retrieval.” ACM Computing Surveys (CSUR) 45.1 (2012): 1-44.

[6] Alkhatib, Manar, and Khaled Shaalan. ”The key challenges for Arabic machine translation.” Intelligent Natural Language Processing: Trends and Applications. Springer, Cham, 2018. 139-156. https://doi.org/10.1007/978-3-319-67056-0_8

[7] Thanh, Thao Phan Thi. Machine translation of proper names from english and french into vietnamese: an error analysis and some proposed solutions. Diss. Université de Franche-Comté, 2014.

[8] Guzmán, Francisco, et al. ”The flores evaluation datasets for low-resource machine translation: Nepali-english and sinhala-english.” arXiv preprint arXiv:1902.01382 (2019).

[9] Aqlan, Fares, et al. ”Improved Arabic–Chinese machine translation with linguistic input features.” Future Internet 11.1 (2019): 22. https://doi.org/10.3390/fi11010022

[10] Harrat, Salima, Karima Meftouh, and Kamel Smaili. ”Machine translation for Arabic dialects (survey).” Information Processing & Management 56.2 (2019): 262-273. https://doi.org/10.1016/j.ipm.2017.08.003

[11] Finch, Andrew, et al. ”A bayesian model of transliteration and its human evaluation when integrated into a machine translation system.” IEICE transactions on Information and Systems 94.10 (2011): 1889-1900. https://doi.org/10.1587/transinf.e94.d.1889

[12] Salawu, Abiodun, and Asemahagn Aseres. ”Language policy, ideologies, power and the Ethiopian media.” Communicatio 41.1 (2015): 71-89. https://doi.org/10.1080/02500167.2015.1018288

[13] Menuta, Fekede. ”Over-differentiation in Amharic orthography and attitude towards reform.” The Ethiopian Journal of Social Sciences and Language Studies (EJSSLS) 3.1 (2016): 3-32.

[14] Kaur, Kamaljeet, and Parminder Singh. ”Review of machine transliteration techniques.” International Journal of Computer Applications 107.20 (2014).

[15] Karimi, Sarvnaz, Falk Scholer, and Andrew Turpin. ”Machine transliteration survey.” ACM Computing Surveys (CSUR) 43.3 (2011): 1-46. https://doi.org/10.1145/1922649.1922654

[16] Le, Ngoc Tan, and Fatiha Sadat. ”Low-resource machine transliteration using recurrent neural networks of asian languages.” Proceedings of the Seventh Named Entities Workshop. 2018. https://doi.org/10.18653/v1/w18-2414

[17] Deep, Kamal, and Vishal Goyal. ”Development of a Punjabi to English transliteration system.” International Journal of Computer Science and Communication 2.2 (2011): 521-526.

[18] Goyal, Vishal, and Gurpreet Singh Lehal. ”Hindi-Punjabi Machine Transliteration System (For Machine Translation System).” George Ronchi Foundation Journal, Italy 64.1 (2009): 2009. https://doi.org/10.4304/jetwi.2.2.148-151

[19] Duan, Xiangyu, et al. ”Report of NEWS 2016 machine transliteration shared task.” Proceedings of the Sixth Named Entity Workshop. 2016.https://doi.org/10.18653/v1/w16-2709

[20] Finch, Andrew, and Eiichiro Sumita. ”Transliteration using a phrase-based statistical machine translation system to re-score the output of a joint multigram model.” Proceedings of the 2010 Named Entities Workshop. 2010. https://doi.org/10.3115/1699705.1699719

[21] Ngo, Hoang Gia, et al. ”Phonology-augmented statistical transliteration for low-resource languages.” Sixteenth Annual Conference of the International Speech Communication Association. 2015.

[22] Laurent, Antoine, Paul Deléglise, and Sylvain Meignier. ”Grapheme to phoneme conversion using an SMT system.” 10th Annual Conference of the International Speech Communication Association 2009 (INTERSPEECH 2009). 2009. https://doi.org/10.21437/interspeech.2009-243

[23] Finch, Andrew, et al. ”Target-bidirectional neural models for machine transliteration.” Proceedings of the sixth named entity workshop. 2016. https://doi.org/10.18653/v1/w16-2711

[24] Shao, Yan, and Joakim Nivre. ”Applying neural networks to English-Chinese named entity transliteration.” Proceedings of the sixth named entity workshop. 2016. https://doi.org/10.18653/v1/w16-2710

[25] Thu, Ye Kyaw, et al. ”Comparison of grapheme-to-phoneme conversion methods on a myanmar pronunciation dictionary.” Proceedings of the 6th Workshop on South and Southeast Asian Natural Language Processing (WSSANLP2016). 2016.

[26] Yao, Kaisheng, and Geoffrey Zweig. ”Sequence-to-sequence neural net models for grapheme-to-phoneme conversion.” arXiv preprint arXiv:1506.00196 (2015). https://doi.org/10.21437/interspeech.2015-134

[27] Rao, Kanishka, et al. ”Grapheme-to-phoneme conversion using long short-term memory recurrent

neural networks." 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2015. https://doi.org/10.1109/icassp.2015.7178767

[28] Tedla, Tadele. "amLite: Amharic Transliteration Using Key Map Dictionary." arXiv preprint arXiv:1509.04811 (2015).

[29] Gezmu, Andargachew Mekonnen, Andreas Nürnberger, and Tesfaye Bayu Bati. "Neural Machine Translation for Amharic-English Translation." ICAART (1). 2021. https://doi.org/10.5220/0010383905260532

[30] Biadgligne, Yohanens, and Kamel Smaïli. "Parallel Corpora Preparation for English-Amharic Machine Translation." International Work-Conference on Artificial Neural Networks. Springer, Cham, 2021. https://doi.org/10.1007/978-3-030-85030-2_37

[31] Biadgligne, Yohannes, and Kamel Smaïli. "Offline Corpus Augmentation for English-Amharic Machine Translation." ICICT CPS conference proceedings. 2022. https://doi.org/10.1109/icict55905.2022.00030

[32] Yuliani, S. Y., et al. "Hoax news validation using similarity algorithms." Journal of Physics: Conference Series. Vol. 1524. No. 1. IOP Publishing, 2020.

[33] United States Board on Geographic Names, and United States. Defense Mapping Agency. Gazetteer of Ethiopia: Names Approved by the United States Board on Geographic Names. Defense Mapping Agency, 1982. https://doi.org/10.5962/bhl.title.39085

[34] Klein, Guillaume, et al. "Opennmt: Open-source toolkit for neural machine translation." arXiv preprint arXiv:1701.02810 (2017).

[35] Andrabi, Syed Abdul Basit, and Abdul Wahid. "A Comprehensive Study of Machine Translation Tools and Evaluation Metrics." Inventive Systems and Control. Springer, Singapore, 2021. 851-865. https://doi.org/10.1007/978-981-16-1395-1_62

[36] Sennrich, Rico, Barry Haddow, and Alexandra Birch. "Neural machine translation of rare words with subword units." arXiv preprint arXiv:1508.07909 (2015). https://doi.org/10.18653/v1/p16-1162

[37] McCandlish, Sam, et al. "An empirical model of large-batch training." arXiv preprint arXiv:1812.06162 (2018).

[38] Yao, Zhewei, et al. "Large batch size training of neural networks with adversarial training and second-order information." arXiv preprint arXiv:1810.01021 (2018).

[39] Alom, Md Zahangir, et al. "A state-of-the-art survey on deep learning theory and architectures." Electronics 8.3 (2019): 292.

[40] Peris, Álvaro, and Francisco Casacuberta. "NMT-Keras: a very flexible toolkit with a focus on interactive NMT and online learning." arXiv preprint arXiv:1807.03096 (2018). https://doi.org/10.2478/pralin-2018-0010