UNIVERSITY OF CAPE TOWN

DEPARTMENT OF COMPUTER SCIENCE

CS/IT Honours Final Paper 2021

Title: Transformer Neural Machine Translation For Nguni Languages

Author: Deevesh Beegun

Project Abbreviation: SMT-NMT

Supervisor(s): Dr. Jan Buys

Category	*Min*	*Max*	Chosen
Requirement Analysis and Design	*0*	*20*	0
Theoretical Analysis	*0*	*25*	0
Experiment Design and Execution	*0*	*20*	20
System Development and Implementation	*0*	*20*	10
Results, Findings and Conclusions	*10*	*20*	20
Aim Formulation and Background Work	*10*	*15*	10
Quality of Paper Writing and Presentation	*10*		10
Quality of Deliverables	*10*		10
Overall General Project Evaluation (*this section allowed only with motivation letter from supervisor*)	*0*	*10*	0
Total marks		**80**	

Transformer Neural Machine Translation For Nguni Languages

Deevesh Beegun
Department Of Computer Science
University Of Cape Town
Cape Town, South Africa
bgndee001@myuct.ac.za

ABSTRACT

Neural Machine Translation (NMT) has shown significant improvements over traditional phrase-based machine translation over recent years. Nonetheless, NMT models have a steep learning curve with respect to the amount of data thus underperform when the amount of training data is limited, as in the case of low resource languages. South African languages being under-resourced have achieved low performance in the machine translation paradigm. To address this issue, we compare different data augmentation techniques on two Nguni languages, namely, IsiXhosa and IsiZulu, with English to IsiXhosa and English to IsiZulu baseline models. The first data augmentation technique makes use of target-side monolingual data to augment the amount of parallel data via back-translation (convert target side language into source side language) and the second technique involves training a multilingual model on a joint set of bilingual corpora containing both the IsiXhosa and the IsiZulu language. We evaluate each model on the publicly available Autshumato evaluation set[1] based on their BLEU scores and show that both techniques result in an improvement in BLEU scores over the baseline models. Moreover, we find that the first technique slightly outperforms the second technique.

CCS CONCEPTS

• **Computing methodologies** → **Machine translation**.

KEYWORDS

Neural Machine Translation, Low Resource Language, Transformer Architecture

1 INTRODUCTION

Neural Machine Translation (NMT) has become the new state of the art, outperforming traditional phrase-based machine translation in recent years [21] [12]. The Transformer architecture [19] coupled with NMT, which is a widely used approach, has shown significant improvements in machine translation performance when trained on high resource languages [21]. However, NMT models require a large amount of parallel data, which are often sparse in low resource languages, thus under-perform when trained on a limited amount of data. In addition to a large amount of data required, the transformer architecture requires a significant amount of hyper-parameters tuning, especially in low resource settings [3].

South African languages being under-resourced have achieved low performance in the machine translation paradigm. To address this issue, we train different Neural Machine Translation models on two Nguni languages, namely, IsiXhosa and IsiZulu, as target language and English as the source language. The Nguni language is a South African language that forms part of a larger group, namely, the Bantu language. In addition to being low resource languages, Bantu languages have a complex structure due to their agglutinating morphology [22]. As a result, they often have large vocabularies, which makes it hard for Neural Machine Translation models to handle such languages [13]. To make the language more interpretable by the NMT models we use Byte Pair Encoding (BPE) [18] sub word tokenization to break large vocabularies into smaller sub word units.

In this study, we compare the performance of different data augmentation techniques against English to IsiXhosa and English to IsiZulu baseline models. The data augmentation techniques involve training a reverse model to back-translate target-side monolingual data into the source-side language. This back-translated data is combined with the parallel corpus which is used to train an NMT model. We hypothesise that a larger monolingual dataset would result in a significant increase in machine translation performance. The second technique involves training a multilingual model on a joint set of bilingual corpora. This technique makes use of the fact that both languages come from the same language subclass and therefore have similar semantics that would help increase the performance of individual translation. We use the Autshumato Evaluation set to compute the BLEU scores of all these models and compare the performance of the models with each other based on their BLEU scores.

The rest of this paper is presented as follows: in section 2, we give some background about the Neural Machine Translation architecture, automatic evaluation of translation models, subword tokenizer, using monolingual data to create additional parallel data and multilingual models. In section 3, we discuss some work that is related to this study. A summary of the datasets used and how they are pre-processed is shown in section 4. Subsequently, in section 5, we discuss how the experiment was implemented and executed. The results for these experiments follows in section 6 and their corresponding discussions in section 7. Finally, we conclude and provide some discussion about the future work that could extend this study.

2 BACKGROUND

2.1 Neural Machine Translation

Neural Machine Translation (NMT) is a recently introduced paradigm that has achieved state of the art performance outperforming traditional machine translation methods [4]. It models the machine translation process by using a single, large Neural Network that takes as input a sentence and outputs its corresponding translation one element at a time, in an end-to-end fashion. This allows all

[1] Available at: https://repo.sadilar.org/handle/20.500.12185/506

parameters of the model to be simultaneously changed to maximise translation performance.

2.2 Encoder-Decoder Model

Most Neural Network translation relies on the encoder-decoder network [4]. The encoder in an encoder-decoder or sequence-to-sequence network takes an arbitrary length input and produces a fixed-length vector representation of the input, as output. This output is known as the context vector which is used by the decoder to produce a variable-length translation of the input[6].

2.2.1 RNN Encoder-Decoder. This Neural Network architecture was proposed by Cho et al [7]. Given an arbitrary length input sequence $x = (x_1, x_2, ...x_n)$ and another arbitrary length output sequence $y = (y_1, y_2, ...y_n)$, this model learns a conditional distribution over the two sequences as follows:

$$p(y \mid x) = \prod_{n=1}^{N} p(y_n \mid y_n, x) \tag{1}$$

where x is the source text and y the target text.

The RNN Encoder-Decoder consists of two Recurrent Neural Networks which can be simultaneously trained to maximise the conditional log-likelihood:

$$\max_{\theta} \frac{1}{N} \sum_{n=1}^{N} logp_{\theta} \Pr(y \mid x) \tag{2}$$

where θ is the set of parameters to be learn by the model, x the input sequence and y, the output sequence from a training set.

The first RNN, the encoder reads a sequence of inputs $x = (x_1, x_2, ...x_n)$ and updates some hidden states, h_t at time t according to the following equation:

$$h_t = \sigma(h_{<t-1>}, x_n) \tag{3}$$

where σ is a non-linear activation function.

After reading the input sequence, the last hidden state represents the context vector, c of the whole input sequence.

The second RNN, the decoder uses the context vector, c as its initial hidden state. It generates an output sequence by predicting y_t in a hidden state, h_t. y_t and h_t both depend on the previous hidden state, the previous output y_{t-1} and the context vector, c [7].

This can be formally represented as follows:

$$p(y_t \mid \{y_1, y2, ..., y_{t-1}, c\} = \sigma(y_{t-1}, h_t, c) \tag{4}$$

where σ is a non-linear activation function.

2.2.2 Attention mechanism. With the encoder and decoder separated, the decoder only knows about the source text through the context vector. The Neural Network must compress all the information from the source text into the final state of the encoder. This makes the Neural Network unable to give correct translations for longer sentences [4].

The attention mechanism proposed by Bahdaunau et al [4]. solves this issue by making all information from all the hidden states in the encoder available to the decoder instead of only the last hidden state. This is achieved by taking a weighted sum of all the hidden states in the encoder and creating a fixed-length vector from the weighted sum. This fixed-length vector is then used by the decoder. The weighted sum changes depending on the current token that is being processed by the decoder thus making the context vector dynamic [4].

2.2.3 Transformers. While Recurrent Neural Network and other sequence-to-sequence models have been the state of the art in the machine translation paradigm, their inherent sequential nature prevents them from being parallelised [20]. This causes their performance to degrade over long sentences. A new model architecture, transformers, proposed by Vaswani et al [20]. rely solely on attention mechanism allows more parallelism.

The transformer has a similar overall architecture to neural sequence models with an encoder that compresses an input sequence to a fixed-length vector and the decoder uses this fixed-length vector to produce an output sequence one element at a time. However, the transformer uses stacked, point-wise, fully connected layers for both the encoder and decoder. The encoder consists of six stacked identical layers with each layer containing two sublayers. The first layer is a multi-head self-attention mechanism, and the second layer is a fully connected feed-forward network. These two sublayers consist of a residual connection around them, followed by a layer normalization. The decoder contains similar layers and connections as the encoder with the addition of a multi-head attention layer which performs multi-head attention on the output of the encoder. Since the decoder is auto-regressive the self-attention layer stack is also modified to prevent the decoder from being conditioned on future words.

The attention function in the transformer takes as input a query and key-value pairs and returns the weighted sum of the values [20]. A compatibility function is used to compute how much weight to assign to each value. The transformer uses the Scaled Dot-product Attention where the dot product of the query of dimension, d_q is computed with the keys of dimension,d_k. Which is then divided by $\sqrt{d_k}$ and a softmax function is applied to obtain the weights on the values.

2.3 Automatic Evaluation: BLEU

The Bilingual Evaluation Understudy (BLEU) method of evaluating translation was proposed by Papineri et al [17]. It is fast, inexpensive, provides an objective view and strongly correlates to human evaluation [9]. It works by comparing n-grams (sequence of n words) of a machine translated output with n-grams of an equivalent human translated text and count the number of matches between them. This represents the precision measure, p_n which is modified to eliminate repetitions. Otherwise, over-generated words by the machine translation would result in absurd but high precision measurements [17]. This can be generalised for a multiple sentence test corpus as follows:

$$P_n = \frac{\sum_{S \in C} \sum_{ngram \in S} Count_{match}(ngram)}{\sum_{S \in C} \sum_{ngram \in C} Count(ngram)} \tag{5}$$

where S is the machine translation output, C is the complete test corpus.

To prevent sentences that are too short from getting a high precision, a Brevity penalty (BP) can be added over the corpus.

BP can be calculated as follows:

$$BP = \begin{cases} 1 & c > r \\ e^{\left(1-\frac{r}{c}\right)} & c \leq r \end{cases} \tag{6}$$

where r is length of the human translated text and c the length of the machine translated text.

Equation (2) outputs a 1 if the c is greater than r. Otherwise a penalty factor of $e^{\left(1-\frac{x}{y}\right)}$ is applied.

The BP can then be applied to the BLEU score as follows:

$$BLEU = BP \times \left(\prod_{n=1}^{4} p_n \right) \tag{7}$$

2.4 Subword Segmentation

Bantu languages have a rich noun class system and a complex structure due to their agglutinating morphology [22]. This makes it hard for Neural Machine Translation models to handle such languages [13] and requires mechanisms to go below word-level [18]. As such, in this study we make use of Byte Pair Encoding [18] sub-word tokenization to break large vocabularies into smaller subword units which can be easily interpreted by Neural Translation models. In addition, this help translates rare words more accurately and generate words that were not seen during training.

This technique of using subword units to encode rare words was proposed by Sennrich et al. [18]. Their main goal was to enable neural machine translation models to generate translations for words that were not present in the training set but instead to use known subword units when translating unknown words.

2.5 Using Monolingual Data

The performance of a Machine Translation model depends on the amount of data the model has been trained on; consequently, the more data we have, the better our model will perform. While phrase-based Machine Translation models like Statistical Machine Translation have benefited by using monolingual data as training data, Neural Machine Translation uses only parallel data for training which are often sparse, especially for low resource languages [18]. On the other hand, in some cases, there is a substantial amount of monolingual data available for the target language.

Sennrich et al. [18] showed that monolingual training data can be treated as additional parallel training data which could improve the quality of translation models by mixing monolingual target sentences into the training set. They proposed two techniques to achieve this: the first one treats monolingual training examples as parallel examples with an empty source side. The second technique that the author proposed is to pair monolingual training instances with a synthetic source sentence. This synthetic data is obtained by using a reverse model to translate the monolingual target text into the source language. This process is known as back-translation. In this study, we will use the second technique since it resulted in a greater improvement in the BLEU score compared to the first technique.

2.6 Multilingual Translation

The performance of Machine Translation models can be improved by training the models on a joint set of bilingual corpora with languages that have similar semantics [8]. This brings multilinguality which helps improve individual translations [10].

In this study, we will make use of the technique proposed by Dong et al [8] which uses a one-to-many translation model to improve translation performance. More specifically, we will train a machine translation model with English as the source language and both IsiXhosa and IsiZulu as target languages, in an attempt to alleviate the data sparsity problem of these two languages. We hypothesize that this will help increase the translation quality of English to IsiZulu by taking advantage of the semantic similarity of these two languages as well as the additional parallel corpora for the translation from English to IsiXhosa.

3 RELATED WORK

Little research has been conducted in the machine translation field for South African Languages. However, several bench-marking using the state of the art Neural Machine Translation has been conducted to provide some groundwork in this particular field [1] [15] [14].

One of those benchmarking studies performed by Martinus et al [1] compared the performance of convolution sequence to sequence and transformer architecture on five different South African languages. They showed that the Transformer model outperformed the convolution sequence to sequence model on all those languages, with Afrikaans achieving the highest BLEU scores despite having the smallest corpus size. IsiZulu however, had a BLEU score of 1.34 which was the lowest among all the other languages. The authors attributed this bad performance to the agglutinating nature of the language as well as the size and quality of the data.

Another study conducted by Nyoni et al [16] compared zero-shot learning, transfer learning and multilingual learning on three Bantu languages, namely, Shona, IsiXhosa and IsiZulu. They achieved an 18.6 ± 0.1 BLEU score with their multilingual English to IsiXhosa to IsiZulu model, yielding a gain of 9.9 over their baseline model. They showed that multilingual learning outperformed both transfer learning and zero-shot learning.

To the best of our knowledge, no studies have made use of synthetic parallel sentences obtained from monolingual data to train Neural Machine Translation models for South African Languages. Nonetheless, this technique has been widely used to improve the translation performance of other non-South African languages [18] [23] [5].

Sennrich et al. [18] showed that monolingual training data can be treated as additional parallel training data which could improve the quality of NMT systems by mixing synthetic parallel sentences obtained from target side monolingual data into the training set. This synthetic data is obtained by translating the monolingual target text into the source language by training a reverse model that back-translates from the target language to the source language. An NMT

Table 1: Summary of the number of sentences in each dataset used to train our models. A subset of the c4 multilingual dataset was used due to its massive size. The MeMaT dataset contained smaller datasets that were combined into a single corpus.

Datasets	Number of Sentences	
	IsiXhosa	IsiZulu
SADiLaR (parallel)	126708	35489
SADiLaR (monolingual)	233192	-
JW300 (parallel)	866748	1046572
C4 (monolingual)	597242 (subset)	623981 (subset)
MeMaT	446065 (combined)	-

model is then trained on the augmented parallel corpora to increase the performance of the translation models.

In this study, we will use the technique proposed by Sennrich et al [18] to improve the translation performance for two South African languages namely, IsiXhosa and IsiZulu by using their respective monolingual data to create additional parallel data.

4 DATASETS

Monolingual corpora consisting of IsiZulu and IsiXhosa sentences and bilingual corpora consisting of aligned translation sentences in separate text files were retrieved from publicly available sources. These sources include the South African Center for Digital Language Resources (SADiLaR)[2] where aligned parallel corpora containing translation for English to IsiXhosa and English to IsiZulu and monolingual corpora containing IsiXhosa sentences were obtained. English to IsiXhosa and English to IsiZulu parallel corpora were extracted from the JW300 parallel corpus [2] which was retrieved from the Opus Corpus website[3]. This corpus contains over 300 sentences for different languages and is originally stored in XML files which were converted into plain text using the opus tools[4]. Additional parallel corpora containing translated sentences from English to IsiXhosa were retrieved from an online repository which has been made available as a result of the Medical Machine Translation project (MeMaT)[5]. These datasets were combined into a single corpus. In addition, a subset of the c4 multilingual dataset[6] containing monolingual IsiXhosa and IsiZulu corpus was used. To evaluate our models we used the Autshumato Machine Translation Evaluation set [7] which consists of 500 sentences for every official South African language. These sentences have been translated separately by four different professional human translators. Table 1 provides a summary of the above datasets.

All datasets from the different sources have been acknowledged and all copyrights have been taken into consideration.

4.1 Data pre-processing

The data pre-processing steps involved combining the parallel corpora for each language into a single corpus, removing empty lines in both the target and the source language corpus in the case of parallel texts, removing extra space between words and removing sentences that are smaller than five words and greater than 200 words. The number of words was chosen to remove bad quality sentences while at the same time preserving the quantity of data for training the models. In the case of the monolingual texts, duplicate sentences were removed to reduce similar sentences in the test and training set thus preventing data leakage. However, in the case of the parallel texts, deduplication was not performed to preserve the alignment of the target and source sentences. These processes were done by using a pre-written cleaning script obtained from the Moses library[8].

After the cleaning process, the data were randomised and partitioned into 80 to 20 training and testing sets. A validation set was derived from the training set by appending sentences with a line number that are divisible by 100 (modulus 100) into a validation text file.

Due to the agglutinating morphology of Nguni languages, they have a complex structure which is often difficult for Neural Machine translation models to handle [9]. We used Byte Pair Encoding (BPE) subword tokenization implemented by Moses [9] to break large vocabularies into smaller subwords and to decompose rare words into meaningful subwords instead of being replaced by an unknown token. This enables the models to process unseen words and handle large vocabularies more efficiently.

To apply BPE, we used a tokenizer to tokenize the dataset set for both the target and the source language. The tokenized training set for both languages were then combined into a single text file which was used to train a BPE model (jointly learn). This model was then used to apply BPE to the entire dataset. The vocabulary size of the BPE model was varied on a range from 2000 to 30000 to get the best performing model.

5 DESIGN AND EXPERIMENTS

The Fairseq modelling Toolkit[10] which is written in Pytorch was used to implement the different Transformer Neural Machine Translation models used in this study. These models were initially trained using Google Colab[11] on a combination of GPUs consisting mainly of the Nvidia k80s, T4s, P4s and P100s. In addition to the previously mentioned GPUs, the Center for High Performing Computing (CHPC) cluster[12] was used to significantly reduce the training time of the models.

All Neural Machine Translation models in this study used a variant of the transformer architecture from [19] consisting of 6 encoder and decoder layers, 8 attention heads, Feed-Forward Networks of dimension 2048 and embedding layers producing an

[2]Available at: https://repo.sadilar.org/handle/20.500.12185/1
[3]Available at: https://opus.nlpl.eu/
[4]Opus Tool can be downloaded from: https://opus-codec.org/downloads/
[5]Memat dataset retrieved from: https://github.com/mkeet/MeMaT
[6]Can be downloaded from: https://github.com/allenai/allennlp/discussions/5265
[7]Available at: https://repo.sadilar.org/handle/20.500.12185/506

[8]https://github.com/moses-smt/mosesdecoder/blob/master/scripts/training/clean-corpus-n.perl
[9]Available at: https://github.com/moses-smt/mosesdecoder/blob/master/scripts/training/clean-corpus-n.perl
[10]Downloaded from: https://github.com/pytorch/fairseq
[11]https://colab.research.google.com/notebooks/intro.ipynb?utm$_source = scs-index$
[12]https://www.chpc.ac.za/

output of dimension $d_{model} = 512$. In addition, the models used a rectified linear activation function (ReLU).

5.1 Model training and tuning

Each model was trained for 15 epochs and patience of 5 was used to stop training if valid performance did not improve for 5 consecutive runs.

5.1.1 Baseline model. English to IsiXhosa and English to IsiZulu baseline models were trained with different hyper-parameters to get the best performing model. These hyper-parameters include the regularization parameters namely, dropout and weight decay to prevent the model from overfitting the training data. These parameters were tested on a range from 0.1 to 0.3 for dropout and from 0 to 0.1 for weight decay. In addition, label smoothing [9] of value 0.1 was applied to prevent overfitting and improve generalization. This value was kept constant throughout all the subsequent models. Following [19] we optimized the models using the Adam Optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.98$ and $\epsilon = 10^{-9}$. using the following formula:

$$Lrate = d_{model}^{-0.5}.min(step\_num^{-0.5}, step\_num).warmup\_steps^{-1.5} \quad (8)$$

with a warmup step of 4000, the learning rate was varied by linearly increasing the learning rate for the first 4000 steps and then decreasing it proportionally to the inverse square root of the step number.

The above hyper-parameters for our best performing baseline models were used for all of our subsequent models.

5.1.2 Model trained on augmented parallel data. To convert the target side monolingual data into additional parallel data a reverse machine translation model, that translates the target side language into the source side, was trained to back translate the target language into the source language. This was done by combining the monolingual data obtained from the different sources into a single corpus for each language and then splitting the combined data into 25 shards with each shard containing the same amount of data except for the last shard. These shards were binarized and back-translation was performed over these binarized shards which were combined into one corpus. Back translated sentences were then extracted using a pre-written python script (extract_bt_data.py), provided by the Fairseq Toolkit, from these back-translated combined shards. A length ratio filter was applied on these back-translated shards to filter sentences that are smaller than five words and greater than 200 words. This filtered data was then combined with the parallel data. These steps were repeated for each language to convert their respective monolingual data into additional parallel data.

A Neural Machine Translation model was trained on these augmented parallel data with 600000 monolingual sentences. To compare the effect of the amount of monolingual data used on the performance of the models, different subsample size of the monolingual dataset was used. The subsample size was chosen on a range from 0 (baseline) to 600000 sentences. The BLEU scores for each model were recorded in table 5 and table 6 for IsiXhosa and IsiZulu respectively.

Table 2: Shows the BLEU scores obtained for the Baseline models as well as for the different models obtained using the different data augmentation techniques. All these models were evaluated on the Autshumato Evaluation set.

	Autshumato dataset	
Model type	IsiXhosa	IsiZulu
Baseline	3.80	4.25
Augmented Data	3.94	3.87
Multilingual	3.92	4.24

5.1.3 Multilingual Model. A multilingual model with English as the source language and IsiXhosa and IsiZulu as target languages was trained and some hyper-parameter tuning was performed to get the best performing model. The training steps consisted of training a joint BPE vocabulary model on all three languages and binarizing the languages for both target languages. In addition, following [11] we add an artificial language token at the beginning of the input sentence to specify the target language. This was achieved by using Fairseq decode-langtok command when training the models. This process allowed multilingual translation without the need to modify the Neural Machine Translation architecture.

The encoder for the source language was shared and different decoders were used for the target languages. By sharing the encoder, the model makes full use of the source language corpora and learn semantic and structured predictive representations[8].

5.1.4 Model evaluation. To compute the BLEU scores for each model we combined the Autshumato Evaluation set text files, translated by four different translators, into one text file for both the source and target language. The best checkpoint for each model was then used to compute the BLEU scores for each model. This was done by using the Fairseq interactive command in a non-interactive way, by reading inputs from the text file and outputting the respective translations in a separate text file. The Fairseq score command which uses a 4-gram BLEU was used to calculate the BLEU score of the translated sentences in the output text file.

The model with the best BLEU scores was then compared with each other. The discussion for the comparison is provided in section 6.4

6 RESULTS

In this section, we describe and compare the different results obtained from our experiments in section 5. In section 6.1 we discuss the results obtained from training the English to IsiXhosa and English to IsiZulu baseline models. Following this, in section 6.2 we discuss the results obtained from training models on augmented parallel corpora. Subsequently, in section 6.3 we provide some discussion about the results obtained from training a multilingual model with English as the source language and IsiXhosa and IsiZulu as target languages. Finally, in section 6.4 we compare the different data augmentation techniques based on their respective BLEU scores and provide some discussion. Our experiment findings are summarised in table 2

6.1 Baseline model

The hyper-parameters for the baseline models were independently tuned and the resulting BLEU scores were recorded in table 3 and table 4 for English to IsiXhosa and English to IsiZulu respectively. A dropout of 0.1, weight decay of 0 and a BPE token size of 10000 was found to give the best BLEU scores for the English to IsiXhosa model. For the translation from English to IsiZulu, a dropout of 0.2, weight decay of 0 and a BPE token size of 10000 resulted in the best BLEU scores.

The English to IsiZulu baseline model achieved a BLEU score of 4.25 which was the highest between the two baseline models despite being trained on a smaller dataset. The model translating from English to IsiXhosa which was trained on a larger dataset, however, achieved a lower BLEU score. This high BLEU score for the translation from English to IsiZulu was found to be a result of the model overfitting the training data. In addition, the datasets on which the model was trained consisted of the Autshumato training set, thus may contain a similar type of content as the evaluation set.

Increasing the size of the Byte Pair Encoding token size in both languages resulted in a decrease in BLEU scores. This is due to the models being trained on a small dataset.

6.2 Model trained on augmented parallel data

The English to IsiXhosa model trained on augmented parallel data achieved a BLEU score of 3.94 and the English to IsiZulu model trained on augmented parallel data achieved a BLEU score of 3.87. The former model achieved a higher BLEU score since it was trained on a larger parallel dataset.

An increase in the subsample size of the monolingual data which is used as additional parallel data by back translating the target language into the source language is seen to cause an increase in BLEU scores as expected for both the models. The English to isiXhosa model is seen to perform only slightly better than the English to IsiZulu model when trained on a smaller subsample size. A summary of the BLEU scores for the models trained on the different subsample sizes is provided in table 5 and table 6 for English to IsiXhosa and English to IsiZulu respectively.

6.3 Multilingual model

The English to IsiXhosa multilingual model achieved a BLEU score of 3.92 and the English to IsiZulu multilingual model achieved a BLEU score of 4.24. From these results, it appears that the translation from English to IsiZulu has benefited from the parallel corpora of English to IsiXhosa. This is due to the semantic similarities between the two languages.

6.4 Comparison between models

The English to IsiXhosa model trained on augmented parallel data gained a 0.14 increase in BLEU scores over the baseline English to IsiXhosa model. The English to IsiZulu model did not show any improvement over the baseline model. However, as mentioned in section 6.1 this model is seen to have overfitted the training data. When the model is trained on a larger parallel dataset obtained as a result of the back-translated monolingual data, it takes longer for the model to overfit the training data. This could have resulted in an increase in BLEU scores over the baseline model if a different evaluation set was used to evaluate the baseline model and if the model was trained on a larger dataset.

Using a small subsample size of the monolingual data resulted in the machine translation model trained on augmented parallel data performing worse than the baseline models. However, as the subsample size is increased the performance of the models increase as well.

The English to IsiXhosa and IsiZulu multilingual model gained a 0.12 increase in BLEU scores over the baseline English to IsiXhosa model.

The English to IsiXhosa model trained on augmented parallel corpora outperformed the multilingual model with a BLEU score of 0.02. The Multilingual model for translation from English to IsiZulu outperformed the augmented parallel corpora by 0.37 BLEU points.

7 CONCLUSIONS AND FUTURE WORK

In this study, we used the state of the art Transformer Neural Machine Translation to compare the translation performance of baseline models with models trained using two different data augmentation techniques on low resource South African languages namely, IsiXhosa and IsiZulu. The first technique involved generating additional parallel data from monolingual data via back-translation and the second technique involved training a multilingual model on a joint set of bilingual corpora. We found that both techniques resulted in higher BLEU scores compared to the baseline models. We also found that the model trained on augmented parallel data outperformed the multilingual model for both language pairs. For the multilingual model, we saw a greater improvement for the translation from English to IsiZulu and found that training models with target languages having similar semantics increase translation performance for the language pair with the smallest dataset size. In addition, we investigated how the size of the Byte Pair Encoding token size affects the translation performance of the models and the result showed that both languages performed better on smaller BPE token sizes. We attribute this low performance when using a larger BPE token size as a result of using a small size dataset.

By using publicly available datasets and toolkits we have shown that using data augmentation techniques resulted in an increase in machine translation performance for low resource South African languages. To this end, future work may involve investigating the performance of machine translation on an even larger amount of monolingual data, to create additional parallel data, such as making use of the full c4 multilingual dataset instead of a subset of the dataset. In addition, more hyperparameter tuning for both the baseline and multilingual models could be done and different sizes of BPE token size especially on the lower range could be used. Human evaluation could also be performed in addition to using BLEU scores to compare the evaluate the performance of translation models.

REFERENCES

[1] Abbott, J., and Martinus, L. Benchmarking neural machine translation for southern african languages. In *Proceedings of the 2019 Workshop on Widening NLP* (2019), pp. 98–101.

[2] Agic, Ž., and Vulic, I. Jw300: A wide-coverage parallel corpus for low-resource languages.

[3] Araabi, A., and Monz, C. Optimizing transformer for low-resource neural machine translation. *arXiv preprint arXiv:2011.02266* (2020).

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

[5] Burlot, F., and Yvon, F. Using monolingual data in neural machine translation: a systematic study. *arXiv preprint arXiv:1903.11437* (2019).

[6] Cho, K., Van Merriënboer, B., Bahdanau, D., and Bengio, Y. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259* (2014).

[7] Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078* (2014).

[8] Dong, D., Wu, H., He, W., Yu, D., and Wang, H. Multi-task learning for multiple language translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)* (2015), pp. 1723–1732.

[9] Escribe, M. Human evaluation of neural machine translation: The case of deep learning. In *Proceedings of the Human-Informed Translation and Interpreting Technology Workshop (HiT-IT 2019)* (2019), pp. 36–46.

[10] Ha, T.-L., Niehues, J., and Waibel, A. Toward multilingual neural machine translation with universal encoder and decoder. *arXiv preprint arXiv:1611.04798* (2016).

[11] Johnson, M., Schuster, M., Le, Q. V., Krikun, M., Wu, Y., Chen, Z., Thorat, N., Viégas, F., Wattenberg, M., Corrado, G., et al. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics 5* (2017), 339–351.

[12] Koehn, P., and Knowles, R. Six challenges for neural machine translation. *arXiv preprint arXiv:1706.03872* (2017).

[13] Long, Z., Kimura, R., Utsuro, T., Mitsuhashi, T., and Yamamoto, M. Neural machine translation model with a large vocabulary selected by branching entropy. *arXiv preprint arXiv:1704.04520* (2017).

[14] Martinus, L., and Abbott, J. Z. A focus on neural machine translation for african languages. *arXiv preprint arXiv:1906.05685* (2019).

[15] Martinus, L., Webster, J., Moonsamy, J., Jnr, M. S., Moosa, R., and Fairon, R. Neural machine translation for south africa's official languages. *arXiv preprint arXiv:2005.06609* (2020).

[16] Nyoni, E., and Bassett, B. A. Low-resource neural machine translation for southern african languages. *arXiv preprint arXiv:2104.00366* (2021).

[17] Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics* (2002), pp. 311–318.

[18] Sennrich, R., Haddow, B., and Birch, A. Improving neural machine translation models with monolingual data. *arXiv preprint arXiv:1511.06709* (2015).

[19] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. In *Advances in neural information processing systems* (2017), pp. 5998–6008.

[20] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. Attention is all you need. *arXiv preprint arXiv:1706.03762* (2017).

[21] Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., et al. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144* (2016).

[22] Zerbian, S. A first approach to information structuring in xitsonga/xichangana. *Research in African Languages and Linguistics 7*, 2005-2006 (2007), 1–22.

[23] Zhang, J., and Zong, C. Exploiting source-side monolingual data in neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing* (2016), pp. 1535–1545.

APPENDIX - SUPPLEMENTARY INFORMATION

Table 3: Shows the BLEU scores for the English to IsiXhosa baseline model. Increasing the BPE token size causes a decrease in BLEU scores

dropout	BPE token size	BLEU score
0	2000	3.20
0.1	10000	3.80
0.2	10000	3.60
0	10000	3.35
0	15000	3.41
0	20000	3.32
0	30000	3.29

Table 4: Shows the BLEU scores for the English to IsiZulu baseline model. Increasing the BPE token size causes a decrease in BLEU scores

dropout	BPE token size	BLEU score
0	2000	4.17
0.2	2000	3.85
0.3	2000	3.94
0.2	10000	4.25
0.1	10000	4.20
0	15000	4.18
0	20000	4.08
0	30000	3.51

Table 5: Shows the BLEU scores for the English to IsiXhosa model trained on augmented parallel corpora. An increase in the size of the monolingual data causes an increase in BLEU scores

dropout	subsample size	BPE token size	BLEU scores
0.1	60000	10000	3.44
0.1	250000	10000	3.69
0.1	600000	10000	3.94

Table 6: Shows the BLEU scores for the English to IsiZulu model trained on augmented parallel corpora. An increase in the size of the monolingual data causes an increase in BLEU scores

dropout	subsample size	BPE token size	BLEU scores
0.2	60000	10000	3.43
0.2	250000	10000	3.61
0.2	600000	10000	3.87

Table 7: Shows the BLEU scores for English to IsiXhosa and IsiZulu multilingual model trained with a dropout of 0.1 and 0.2 respectively.

dropout	BPE token size	BLEU scores
0.1	10000	3.92
0.2	10000	3.83

Table 8: Shows the BLEU scores for English to IsiXhosa and IsiZulu multilingual model trained with a dropout of 0.1 and 0.2 respectively.

dropout	BPE token size	BLEU scores
0.1	10000	4.19
0.2	10000	4.24