English↔Twi parallel-aligned Bible Corpus for encoder-decoder based machine translation

Accepted 18th September, 2020

ABSTRACT

This research presents the first work on the deep natural language processing (DNLP) approach for translation purposes from English-Twi (a popular Ghanaian language) language. The paper reports machine translation (MT) experiments on English-Twi parallel Bible corpus with the state-of-the-art neural transformer, a self-attention encoder-decoder. This is the first attempt at engaging MT on English-Twi corpora with about 124K parallel-aligned sentences. We investigated the dynamics of layers and the number of models to fit this language pair's demands through the experiment. Furthermore, we leveraged the maximum posterior-based decoding and performed the n-best beam search on the test data. However, with no baseline translation system, we engaged the Moses toolkit, a Statistical MT (SMT), and the well-known sequence to sequence recurrent neural network (RNN) to serve as a baseline for comparison against the neural transformer. The results were recorded both in BLEU and TER on a newly released evaluation set. This work's findings present the possibility of conveniently adding one of the most important directions for MT's future in these language pairs.

Key words: Akan, beam search, English-Twi corpora, maximum posterior, transformer.

INTRODUCTION

Using English as a mode of teaching and learning instead of one's mother-tongue affects students' ability to learn and perform directly (Kamwangamalu, 2008; Webb, 2002). There is a need for people worldwide to use their language to understand better, especially when using computers or accessing the information on the Internet. Therefore, having a high-quality, English-Twi MT system is necessary. While this appears to be an easy task, it requires various applications, including software like local language spell-checkers, word processors, machine translation systems and search engines. Simultaneously, the amount of work required to develop all NLP aspects for a new language is enormous. Solving the problem with corpus statistical and neural techniques is a rapidly growing field leading to better translation. NMT is an end-to-end approach for automated machine translation (Jean et al., 2015; Luong et al., 2015; T. Luong et al., 2015) which emerges as an up to date problem. It serves as a successful method to encode a variable-length source representation with RNN to a fixed-length vector. And then decode the translation left-to-right with another RNN from the vector representation to variable-length target sequence using a soft-attention mechanism (Mahata, et al., 2018). The attention mechanism layer serves as the interface that bridges the encoder and the decoder.

It decides on frames within encoder representation that the decoder should pay attention to predict the next sub-word unit (Bahuleyan et al., 2018; Michael et al., 2019; Prabhavalkar et al., 2017; Sennrich et al., 2016c). However, due to its inability to maintain a hidden state of the entire past that prevents parallel computation, the Transformer (Vaswani et al., 2017) came to play, to reduce sequential computation. Due to the ubiquitous nature of attention in modern deep learning, the Transformer model architecture
avoids recurrence. Instead, it relies entirely on attention to boost the speed with which these models can be trained. Not only that, but it also draws global dependencies between the encoder and the decoder. This significantly allows more parallelization and achieves a new single-model state-of-the-art BLEU on MT tasks (Vaswani et al., 2017). It also outperforms the Google NMT model (Wu et al., 2016) in specific tasks. The exceptional performance of the Transformer model has led us to focus on the architecture of the sequence-to-sequence (seq2seq) attention-based model as our base model. We together leverage the maximum posterior based decoding and beam search on the English-Twi MT tasks.

Even though the Transformer has gained popularity in the research domain, MT is still far from human-like translations for most of the less famous language pairs, especially languages in the African community. Therefore, we need many research studies and efforts to achieve more effective translation systems. We established a language processing tool for Ghana's most popular language by engaging the neural machine translation model for translation based on English-Twi bilingual parallel corpora. No efforts have been made on for the English-Twi language pair. Although English is a famous language and Twi is the most common language spoken in Ghana. Investigating the English-Twi language pair, however, needs to address existing limitations. First and foremost, the most critical limitation is the number of accessible corpora to train the model. The performance of any machine learning algorithm relies on the quantity of the availability of data. Hence MT is not far from that since a large amount of parallel corpora is required to tune parameters and learn the correspondence between input and output sentences for better performance. As a low-resource language pair, there are no freely available parallel corpora for the English-Twi language pair neither the availability of previous studies nor baseline models to be improved.

**Research goal and contributions**

The amount of work required to develop all aspects of NLP for a new language is enormous as far as MT is concerned. To overcome these problems, automated translation using computers is a good option. The MT system has already solved this problem to some extent, but not entirely. MT systems started operating with the lowest quality and accuracy of the translation. However, it has attracted many researchers to find new translation methods and improve these systems' translation quality due to its influence. With the recent success in MT using neural networks, MT is still far from being a human-like translation. Because of the gap between MT and the English-Twi, this research aims at:

i. Engaging a bilingual English-Twi Bible Corpus, which are the first corpora in these language pairs with parallel-aligned sentences to train MT models for translation.

ii. English-Twi translation with the neural Transformer combined with the maximum posterior based decoding for n-best beam search delivered the best performance.

iii. Ultimate pruning to investigate the dynamics of layers and the number of models to fit this language pair's demands.

iv. With no baseline on this language pair, we engaged a famous SMT toolkit (Moses) and the well-known seq2seq RNN to serve as a baseline for comparison against the neural Transformer.

v. Finally, this work’s findings present the possibility of conveniently adding one of the most important directions for MT’s future in these language pairs.

vi. With no baseline translation system, this work’s results and findings are considerable. They serve as a reference line for the future of machine translation for this language pair.

It is often challenging to learn what a "clean" corpus looks like in lower-resource situations, especially where the target corpus is the only sample of the language's parallel text. We, therefore, performed unsupervised measurements on each sentence pair in our previous work (Adjeisah et al., 2020) for corpus construction. We engage the squared Mahalanobis distances that predicted parallelism on the dataset.

**RELATED WORK**

Seq2seq attention-based encoder-decoder model (Bahdanau et al., 2015; Luong et al., 2015) made of multi-RNN layers emerged dominant paradigm in MT some years back. It achieved a state-of-the-art (SOTA) performance in the translation of language pairs with a considerable amount of training parallel corpora, such as English-French [Bahdanau et al., 2015] and English-German (Jean et al., 2015). The attention mechanism in Long-Short Term Memory (LSTM) (Hochreiter and Schmidhuber, 1997) and convolutional neural networks (CNN) (Bouwrie, 2006) serves as an improvement to the network. With "attention," the network relies not only on the encoder’s final output (z) but can directly attend to the source sentence. It helps to notice part which of the input sequence the network pays attention to (Bachrach et al., 2017; Vig and Belinkov, 2019) and must change while generating a new word in the output. The method makes it possible for the decoder to capture somewhat global information rather than infer based on one hidden state. However, it is characterized by maintaining a hidden state of the entire past that prevents parallel computation within a sequence. Training is slow, as attention is used in combination with RNN. As a result, it becomes critical at longer sequence lengths, as memory constraints limit batching across examples (Vaswani et al., 2017).

It is fair to say that Vaswani et al. (2017)’s, "Attention is
all you need," has shifted NLP's paradigm. It has successfully achieved tremendous SOTA results on existing benchmarks. With the advent of the neural Transformer, which employs neither convolution nor recurrence, experts have shown significant interest. The integration of an explicit representation of position information is significant since the model is otherwise entirely invariant to sequence order. Therefore, attention-based models have used position encodings or biased attention weights based on distance (Parikh et al., 2016). The approach has demonstrated groundbreaking results in many tasks, such as question answering, natural language inference and paraphrase detection. Shaw et al. (2018) presented an efficient way of incorporating relative position representations in the Transformer's self-attention mechanism. They cast the method as an exceptional case encompassing the Transformer's self-attention mechanism to analyze any two input elements' arbitrary relations.

In conclusion, their findings demonstrated a substantial improvement in translation quality on two machine translation tasks, even when replacing its absolute position encodings entirely. Since then, many authors like Ott et al. (2019), reduced precision and considerable batch training, which speed up training by nearly 5x on a single 8-GPU machine with careful tuning implementation self-attention as a baseline. In the perspective of Vaswani et al. (2017) there has been extensive work to improve models with monolingual data, including language model fusion (Burlot and Yvon, 2019; Gulcehre et al., 2015; Gulcehre et al., 2017), back-translation (Sennrich et al., 2016b) and dual learning (Cheng et al., 2016; He et al., 2016). These methods have different advantages and can be combined to reach high accuracy (Hassan et al., 2018). With the advent of powerful pre-trained representations, trained using some flavor of language modeling objectives such as BERT (Devlin et al., 2019), OpenAI GPT (Alec et al., 2018), and ELMo (Peters et al., 2018), a de facto technique for NLP has surfaced. They have become off-the-shelf models pre-trained on gargantuan amounts of data and fine-tune to any task with some smaller in-domain corpus. Although current research works were that NLP architectures could achieve a few percentage points of improvement with BERT.

Many work lines that are still much underexplored will be crucial to push forward the next iteration of NLP progress. Farhan et al. (2020) proposed two systems for unsupervised dialectal NMT, where they present source dialect in the parallel training corpus. They engaged Dialectal to Standard Language Translation (D2SLT) system and the current Google NMT (GNMT) system. The D2SLT achieved a good BLEU score in the unsupervised setting, which is remarkably high compared to the supervised setting's BLEU score. This study tackles the transformer model's performance on the English-Twi corpus, a newly built corpus language pair for MT task. Furthermore, we leveraged the maximum a posteriori based decoding and performed n-best beam search on the test data and evaluate the models on a newly released evaluation set.

**TRANSFORMER OVERVIEW**

Most seq2seq models are of the same interface, which is the encoder-decoder structure (Bahdanau et al., 2015; Cho et al., 2014; Sutskever et al., 2014). In this case, the encoder serves as a successful method to encode a variable-length source representation \( x = (x_1, ..., x_n) \) with RNN to a fixed-length vector \( z = (z_1, ..., z_n) \). Given \( z \), the decoder then translates \( y = (y_1, ..., y_m) \) left-to-right with another RNN from the vector representation back to variable-length target sequence where both components interface via a soft-attention mechanism (Bradbury and Socher, 2016; Luong et al., 2015; Sennrich et al., 2016a). The Transformer model mimics similar architecture except relying entirely on stack self-attention and position-wise, fully connected layers for both the encoder and decoder (Vaswani et al., 2017). The Transformer is a seq2seq attention-based model relying entirely on self-attention without using RNN or convolutions. It habits multi-head attention to increase the speed with which these models can be trained.

**The encoder-decoder**

The model architecture of the Transformer stacks multi-head attention (MHA) and position-wise, fully connected layers for both the encoder and decoder. The encoding component is a stack of encoders. It is composed of a stack of identical layers (\( N = 6 \)). They are all identical in structure, yet do not share weights with layer broken down into two sub-layers. The encoder's input first flows through MHA. This layer helps the encoder pay attention to other words in the input sentence. It encodes a specific word. The output of the MHA layer is fed to a position-wise, fully connected feed-forward network. The architecture is deployed with residual connections around each of the two sub-layers, followed by a normalization layer. The decoder has all the encoder layers, except a third sub-layer that positively engages the decoder to focus on parts over the encoder stack (Vaswani et al., 2017). The MHA layers in the decoder operate in a slightly different way than the one in the encoder. It only permits the network to attend to more preceding positions in the output sequence by masking future positions.

**Attention details**

They processed the model so that the self-attention permits it to operate at other positions for clues for each word's position in the input sequence. It gives a better encoding for the word. As a result, the self-attention becomes a scheme the Transformer employs to foster the knowledge behind
previous words into the one the model is currently processing. The initial step in the attention function is to generate three vectors from the embedding of each word of the encoder’s input vector. Multiplication of the embedding generates these abstraction vectors; query vector, key vector and value vector by three matrices that get trained during the process. It computes the output as a weighted sum of the values. A compatibility function of the query computes the weight assigned to each value with the corresponding key (Vaswani et al., 2017). Scaled dot-product attention is adopted to calculate the score of each word of the input sentence. It is done by taking the Query vector dot product with the key vector of the particular word the model is scoring. The self-attention for a sequence of vectors \( d_j \) of size \( d_k \) is packed into a matrix \((Q,K,V)\), as:

\[
\text{Attention}(Q,K,V) = \text{softmax} \left( \frac{QK^T}{\sqrt{d_k}} \right) V
\]

(1)

Where the input consists of \( Q \) and \( K \) of dimension \( d_k \) and \( V \) of dimension \( d_v \).

Instead of performing single-attention, Vaswani et al. (2017) further refined the self-attention by adding a the MHA mechanism. It expands the model’s ability to focus on the different positions as well as giving the attention layer multiple representation subspaces. It projects the query, keys, and values \( h \) times with different, learned linear projections to \( d_h d_k \) and \( d_d \) dimensions. An essential attention function is performed in parallel on each projected version of the abstraction vectors, yielding \( d_d \)-dimensional output values. These are concatenated and projected again, resulting in the final values. The equations can be represented as:

\[
\text{MultiHead}(Q,K,V) = \text{Concat}(\text{head}_1,...,\text{head}_h)W^v
\]

(2)

\[
\text{head}_i = \text{Attention}(QW^Q_i, KW^K_i, VW^V_i) \]

(3)

Where projections \( W^Q \in \mathbb{R}^{d_h \times d_k}, W^K \in \mathbb{R}^{d_h \times d_k}, W^V \in \mathbb{R}^{d_h \times d_v} \) and \( W^Q_i \in \mathbb{R}^{d_h \times d_k} \) and \( W^K_i \in \mathbb{R}^{d_h \times d_k} \) and \( W^V_i \in \mathbb{R}^{d_h \times d_v} \) are learnable parameter matrices, \( h \) is the number of heads and \( d_{model} \) is the model dimension. The connotation between every word representation \( d_j \) and any other context word \( \hat{d}_j \) is done via the dot-product between the matrix \((Q,K,V)\) and its transpose \( K^T \), yielding a vector of raw connotation values \( QK^T \). The values are then scaled by the dimension of the word representation vectors \( d_j \), followed by the softmax, where their values add up to 1. The resultant vector of normalized connotation values is finally used to weigh the context words. Another probe of Equation 1 without the matrix \((Q,K,V)\) representation but word representation vectors \( d_j \):

\[
a_{ji} = \frac{1}{d_k} d_j^T \hat{d}_i
\]

(4)

The above-described self-attention is only one step in the self-attention layer at the encoder input sentence time. Nevertheless, there are four more steps that are further stated as follows:

1) An immediate is the combination of self-attention with residual connections that directly passes the word representation through.

\[
\text{self-attention}(d_j) + d_j
\]

(5)

2) Next, is a layer normalization step

\[
\hat{d}_j = \text{layer-normalization} \left( \text{self-attention}(d_j) + d_j \right)
\]

(6)

3) The third step is a standard feed-forward activation function (ReLU).

\[
\text{ReLU} = (W\hat{d}_j + b)
\]

(7)

4) The step is further augmented with residual connections and layer normalization as follows.

\[
\text{layer-normalization} = \left( \text{ReLU} \left( W\hat{d}_j + b \right) \right) + \hat{d}_j
\]

(8)

MHA acts similarly to ensembles of relatively small attentions to allow the model to jointly attend to information from different representation subspaces at different positions, which is beneficial to learn complicated alignments between the encoder and decoder. As stated, the decoder has all layers of the encoder, except between them is a third sub-layer that highly engages the decoder to focus on parts over the output sentences of the encoder stack. In total, there are three sub-layers:

- Self-attention: The output words of the encoder are initially encoded by word embedding. At this stage, the self-
attention computation described in Equation 1 is the same. However, the connotation of a word $s_i$ is limited to words sk with, which is the previously generated output words. We express the result of the sub-layer for output word $i$ in the decoder as $\tilde{s}_i$.

- The Attention Mechanism: It mimics the self-attention very closely except that, previously, the model computes self-attention between the hidden states and themselves. At this stage, it computes the attention between the final encoder states $S^\ast$ and the decoder states $S$.

$$Attention(\tilde{S}, \chi) = \text{softmax} \left( \frac{\tilde{S}\chi^T}{\sqrt{d_k}} \right) \chi$$

(9)

Where $\chi$ denotes packed matrix $(Q,K,V)$.

Engaging the same more detailed exposition as Equation (4) for self-attention:

$$a_\mu = \frac{1}{d_k} \tilde{s}_j^T d_i$$  

raw connotation $\left( \frac{\tilde{S}\chi^T}{\sqrt{d_k}} \right)$

$$\alpha_\mu = \frac{\exp(a_\mu)}{\sum_\nu \exp(a_\nu)}$$ normalized connotation(softmax)

self-attention($\tilde{s}_j$)$= \sum_i \alpha_\mu d_i$ weighted sum

(10)

Same as the self-attention layer described above, the attention in the decoder state computation is also augmented by adding residual connections, layer normalization, and an additional activation function (ReLU) layer.

- Feed-forward layer: Lastly, the feed-forward sub-layer is identical to the encoder, that is, $\text{ReLU}(W', \tilde{s}_j + b)$

Individual sub-layers are accompanied by an add-and-norm step of first applying residual connections and then layer normalization. The entire model architecture is shown in Figure 1.

**Posterior beam search based decoding**

We defined the translation problem as a probability of mapping the vector representation $v$, to variable-length target $T$ that maximizes the posterior probability $P(T \mid X)$, given an input $X$ which can be transformed (Kanda et al., 2017) as:

$$\bar{T} = \arg \max_T P(T \mid X)$$

(11)

$$= \arg \max_T \sum_v P(T \mid v) P(v \mid X)^\alpha$$

(12)

$$\approx \arg \max_T P(T \mid v) P(v \mid X)^\alpha$$

(13)
Here, \( P(v | X) \) is a probability from input \( X \) to a fixed-length vector \( v \), \( P(T | v) \) is the probability from fixed-length vector to variable-length target \( T \), and \( \alpha \) is a scaling factor. Length-vector \( v \) and target \( T \) can now transform as a conventional interpolation-based framework using Viterbi-approximation to remove the summation by \( v \) and can be calculated as follows:

\[
P(T | v) = \frac{P(v | T) P(T)}{P(v)^{\alpha}}
\]

Based on equation (14) we propose that both \( P(v | X) \) and \( P(T | v) \) can be considered as seq2seq transformation which can be modeled by attention-based models, targeting the transformer-based self-attention used in this study. This process, which suggests that we perform the 1-best beam search decoder with the transformer is then proposed to directly estimate equation (14). Foremost, the preeminent vector representation sequence is calculated by the Transformer using training data for \( v | X \) and \( T | v \) translations respectively. Foremost, the preeminent vector representation sequence \( v \) is calculated by the Transformer from input \( X \) with beam size. And then, the best word sequence \( T \) is chosen by the Transformer from vector representation sequence to variable-length target with another beam size. The neural Transformer model predicts one output word at a step. We first compute a probability distribution over all the predicted words for the first 5 top choices. We picked the word that received the highest probability as the output word. This method improves the translation accuracy with a little increase as the decoding algorithms for these models keeps a list of the \( n \)-best candidate hypotheses. It expands them and keeps the \( n \)-best expanded hypotheses for prediction, as shown in Table 1. Consider the input sentence: “In the beginning god created the heavens and earth” Output Word Predictions.

### EXPERIMENTAL RESULT

We evaluated the English-Twi translation task approach, a newly built corpus language pair for MT task in both directions. Furthermore, we compare the translation accuracy with RNN and the famous SMT (Moses).

### The English-Twi Corpus

The English-Twi parallel Bible corpus benchmark is a corpus harvested from the You Version Bible website. The Twi corpora contain a single version while the English consist of 4 different versions. The King James Version (KJV), Good News Bible (GNB), Easy to Read Version (ERV) and New International Version (NIV). We stored the 124,400 parallel-aligned in a tab-delimited separation. Verses with the same line number in a line pair are mappings of each other. We present statistics of the released corpus in Table 2 below. The number of words and sentences presented is after sentence-alignment with English-Twi.

### Dataset preprocessing

We used the Natural Language Toolkit (NLTK), a useful tool for loading and cleaning text. It is essential to get data ready for working with machine learning and deep learning algorithms. First, we split the text into sentences and each sentence into words after the dataset’s successful loading. We refined vocabularies by removing words used less than five times in the dataset and replaced it with an unknown token (<UNK>). The next preprocessing was to convert the corpus into lowercase. Finally, we tokenized the datasets with the tokenizer function. We followed by filtering out all standalone punctuation tokens for both languages.

### Training

First, we used 124,400 parallel-aligned sentences for the
Table 2: Statistics of English-Twi Corpus

<table>
<thead>
<tr>
<th></th>
<th>Train</th>
<th>Valid</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>Query</td>
<td>119400</td>
<td>2000</td>
<td>3000</td>
</tr>
<tr>
<td>Max Sentence</td>
<td>Eng</td>
<td>118</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Twi</td>
<td>93</td>
<td></td>
</tr>
<tr>
<td>Ave tokens</td>
<td>Eng</td>
<td>17979</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Twi</td>
<td>23162</td>
<td></td>
</tr>
</tbody>
</table>

English-Twi corpus in our experiment, engaging the Transformer. We appended extra tokens like an unknown token (<UNK>), a padding token (<PAD>), and sentence a start of sentence and end of sentence tokens (<SOS>/<EOS>) to the outputs. We first reshuffled the corpus, used 85% as training, 5% validation and 10% for testing. Hence, 119400 of training samples, 2000 evaluation samples for training. With batch size initialization of 65 and epoch of 30, we had 1874 training batches, which produces 66,100 total steps on both training and evaluation. While training is ongoing, we run validation at every epoch by hypothesizing translation and calculating the BLEU score. We optimize and tune the models on the training data and finally evaluate the models' robustness on the validation sets. It gave a clear intuition of a BLEU score improvement at every epoch stage. For example, at epoch 1, the model recorded illegal division by zero, which means the model did not learn anything at this stage. Therefore there was no length of translation to be compared with a reference length for BLEU score calculation. The model finally appended the BLEU score on the hypothesis translations on subsequent epochs. To limit over fitting and improve the model's ability to generalize, we engaged Srivastava (Nitish et al., 2014) standard Dropout rate. With weight constraints such as the max norm constraint, regularization to avoid over fitting eventually increases translation accuracy.

For test purposes and verification of the effectiveness of the leveraged posterior and beam search decoder used in this study, the test data were converted into word sequences using the trained $P(T|v)$ model. The Transformer calculates vector representation sequence $v$ from input $X$ with beam size. And then, the best word sequence $T$ is chosen by the Transformer from vector representation sequence to variable-length target with another beam size. As mentioned, we first compute a probability distribution over all the predicted words for the first top 5 choices. The word that received the highest probability is finally picked as the output word. When predicting the first word of the output sentence, we keep a beam of the top $n$ most likely word choices. Their probability scores them. We then use each of these words in the beam in the conditioning context for the next word. Due to this conditioning, we make different word predictions for each. We now multiply the partial translation (just the probability for the first word) and the probabilities from its word predictions. We select the highest-scoring word pairs for the next beam. Experimental results showed an improvement in both BLEU score and TER when vector unit sequences calculated by the Transformer from input $X$ to vector unit sequence $v$ were engaged to these corresponding sentence-alignments. For our base model, using the hyperparameters described in attention is all you need (Vaswani et al., 2017), except for the batch size of 64, layer=4, and a Dropout of 0.5 combined with a weight constraint of 0.5 to combat overfitting, each training step took about 1.25 seconds on each step for a total of 66,100 steps. We experimented on the model (d_model=256 and 512) against layer=4 and layer=6 layer parameters to select a suitable one for our problem. Also, we used warm up steps=4000 for our base model.

Evaluate neural translation model

We evaluated the model on both the training and the testing dataset. Ideally, we used a separate validation dataset of 2000 to help with model selection during training instead of the test set. We engaged two metrics to evaluate the translation qualities of the MT systems automatically that is, the Bilingual Evaluation Understudy (BLEU) score (Papineni et al., 2002) and Translation Error Rate (TER) (Snover et al., 2006). The model showed excellent performance on the training dataset and was idealized to perform well on the test set. BLEU is the most common algorithm used to evaluate the quality of MT systems automatically. It computes the translated precision by counting the number of matches between n-grams of a machine-translated sentence and corresponding reference. Similar to BLEU is the TER, a method used by MT specialists to define the amount of post-editing required for translation job work. To compute the TER, we further processed the source and reference corpus on the test set using tercom.jar\(^1\) as it is much faster.

With no baseline translation system on this corpus, we compared the model against the widely-used Moses\(^2\) toolkit (Hoang and Koehn, 2008), an open-source state-of-the-art phrase-based SMT system. We process the corpus by excluding sentences that are longer than 80 words in training. We set the vocabulary size of both source and target languages to be 30K for English↔Twi, which covers

\(^{1}\text{http://www.cs.umd.edu/~snover/tercom/}\)

\(^{2}\text{https://github.com/moses-smt/mosesDecoder}\)
Table 3: Official translation results.

<table>
<thead>
<tr>
<th></th>
<th>BLEU</th>
<th>TER</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMT Moses En-Twi</td>
<td>11.81</td>
<td>77.63</td>
</tr>
<tr>
<td>MNT RNN En-Twi</td>
<td>13.33</td>
<td>75.31</td>
</tr>
<tr>
<td>Transformer</td>
<td>14.68</td>
<td>70.64</td>
</tr>
<tr>
<td>SMT Moses Twi-En</td>
<td>11.53</td>
<td>77.88</td>
</tr>
<tr>
<td>MNT RNN Twi-En</td>
<td>13.35</td>
<td>75.22</td>
</tr>
<tr>
<td>Transformer Twi-En</td>
<td>14.66</td>
<td>70.59</td>
</tr>
</tbody>
</table>

Table 4: Comparison of the transformer as end to end model and the beam decoder with transformer as base model.

<table>
<thead>
<tr>
<th>BLEU SCORE</th>
<th>Transformer (no beam)</th>
<th>Beam Decoder</th>
<th>Δdif</th>
</tr>
</thead>
<tbody>
<tr>
<td>En-Twi</td>
<td>14.49</td>
<td>14.68</td>
<td>+0.19</td>
</tr>
<tr>
<td>Twi-En</td>
<td>14.31</td>
<td>14.66</td>
<td>+0.35</td>
</tr>
</tbody>
</table>

Table 5: Summary of the model parameters configuration and performance of BLEU score.

<table>
<thead>
<tr>
<th>Model parameters configuration</th>
<th>model performance</th>
<th>BLEU score</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>evaluation</td>
</tr>
<tr>
<td>N block 4</td>
<td></td>
<td>14.93</td>
</tr>
<tr>
<td>6</td>
<td></td>
<td>14.82</td>
</tr>
<tr>
<td>8</td>
<td></td>
<td>14.01</td>
</tr>
</tbody>
</table>

85%/90% of the training data source/target side. We also engaged RNN for NMT comparison. The result was recorded both in BLEU and TER. The results indicate that we require more post-editing efforts for the Moses system to exactly match the reference translations as shown in Table 3. To understand the full effects of incorporating the leveraged posterior beam search decoder, we performed an ablation. We engaged the Transformer as an end-to-end translation and the leveraged posterior beam search decoder model for another run. BLEU scores are recorded and compared. Δdif indicates the difference ratio of our model and transformer as an end-to-end model (Table 4).

Most importantly, these translation tasks involve training corpora with different parallel-aligned sentences ranging from 30k to over 120k sentence pairs. It helps us thoroughly examine the leveraged posterior beam search decoder model's ability on different training data sizes. We also investigated the dynamics of the number of layers and models suitable for a small corpus amount. We observed that the layer n = 6 did not improve the translation accuracy for this type of model on the problem resulting in poor performance. The smaller N layer resulted in better performance, with a value of 4. It gave a clear indication that no dataset is the same; hence performing parameter tuning such as layers=4 and model=512 is better on our translation task. Table 5 shows the list of the model parameters configuration and its performance on English↔Twi corpora. We used four layers during training to serve as a base model. The model did not learn general-purpose with six or more layers, hence poor performance on the validation set. With four layers, the model learned well, resulting in better translation on the test set. The English language is not very similar to Twi in terms of grammatical and phrase structure; Figure 2 visualizes the attention matrix. The <SOS> symbol shows the start-of-sentence input, the <EOS> symbol shows the ending-of-sentence output. Moreover, the <PAD> symbol would fill in an empty sequence if the current batch data size is shorter than the time steps. The high-values in the visualization indicate content-dependent heads and low values indicate content-independent (position-based) heads.

Comparison between SMT and NMT

For both translation directions, NMT systems outperform the SMT system in terms of the BLEU and TER scores. The neural Transformer outperforms SMT by 2.87 and 3.13 BLEU points in the baseline system and dropped 6.99 and
7.29 TER points on English↔Twi, respectively. Also, the Transformer outperforms RNN by 1.35 and 1.31 BLEU points and dropped 4.67 and 4.63 TER points. Finally, on RNN against Moses, RNN recorded a difference of 1.52 and 1.82 BLEU points higher and dropped 2.32 and 2.64 TER points. Note that the lower the TER score, the lesser amount of post-editing for translation job work. It is similar to the results of popular language pairs, such as English, Chinese, French, German, Russian and Spanish (Bentivogli et al., 2016; Junczys-Dowmunt et al., 2016) in which NMT is superior to SMT.

Sample translations

Machine translation is perhaps one of the most difficult in artificial intelligence due to the natural ambiguity and suppleness of human language. Because, given a text sequence in a particular language, there is no one distinct preeminent translation of that text to another language. It makes the challenge of automatic machine translation a problematic task. Table 6 shows the sample of translations in both directions. For example, the system translates “medwọ” as “i love you” and “wo ho tesn” as “how are you”. We can also see that the translations were not perfect, with “næbeyeyeæanwümmerennnnnan” translated to “it about four the afternoon” instead of the expected “it was about four in the afternoon”. Also, it duplicates some words during translating from English-to-Twi and vice versa. For example: “this field was near a pond which was full of frogs” was translated as “saabënatakẹ bi aahyəahyəahyəahyə mu mu” instead of “saasareyibënatakẹ bi a mọtorɔahyə mu ma ho”.

CONCLUSIONS AND FUTURE WORK

The study presents the first work on machine translation from English-Twi language pair with the neural Transformer. We further leveraged the maximum posterior-based decoding and performed the n-best beam search on the test data, which yielded tremendous state-of-the-art results. Both BLEU and TER score outperformed the Moses SMT toolkit with about 20% as a baseline for the translation. The results are encouraging. Intuition from varying the parameters is that, for a small dataset using smaller model parameters such as layers = 4 and model = 512 is paramount. An obvious next step is to increase the dataset used to fit the model to about 5000,000 phrases by performing human translation on daily use phrases or creating a system of large English↔Twi sentence-alignment corpus for translation. Another exciting direction would be to improving the low-resource English-Twi parallel-aligned corpus by expanding the training set with synthetic-parallel text.

ACKNOWLEDGMENTS

The author appreciates the Science and Technology
Table 6: Sample translations for each example, we show the source (input), the predicted (predict) and the reference.

<table>
<thead>
<tr>
<th>Twi-to-English translation</th>
<th>Input</th>
<th>Predict</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>&lt;SOS&gt;-ɛyɛaduro&lt;EOS&gt;</td>
<td>it is medicine</td>
<td>It is a medicine</td>
</tr>
<tr>
<td></td>
<td>&lt;SOS&gt;-medɛ wo &lt;EOS&gt;</td>
<td>i love you</td>
<td>I love you</td>
</tr>
<tr>
<td></td>
<td>&lt;SOS&gt;-wo ho tesɛn&lt;EOS&gt;</td>
<td>how are you</td>
<td>How are you</td>
</tr>
<tr>
<td></td>
<td>&lt;SOS&gt;-naɛbeyɛsanwanumɛɛɛnan&lt;EOS&gt;</td>
<td>it about four the afternoon</td>
<td>it was about four in the afternoon</td>
</tr>
<tr>
<td></td>
<td>&lt;SOS&gt;- papa no tedan no anım&lt;EOS&gt;</td>
<td>the man is sitting in front of the house</td>
<td>The man is sitting in front of the house</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>English-to-Twi translation</th>
<th>Input</th>
<th>Predict</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>&lt;SOS&gt; the word became flesh &lt;EOS&gt;</td>
<td>asɛm no bɛyeɛhonam &lt;EOS&gt;</td>
<td>kyɛneawubenya no mu abien</td>
</tr>
<tr>
<td></td>
<td>&lt;SOS&gt; divide the answer that you will get by two &lt;EOS&gt;</td>
<td>kyɛneawubenya no mu abien</td>
<td>kyɛneawubenya no mu abien</td>
</tr>
<tr>
<td></td>
<td>&lt;SOS&gt; i am the voice of one calling in the wilderness &lt;EOS&gt;</td>
<td>meneɛne a mu esse so se &lt;EOS&gt;</td>
<td>meneɛne a ěteɛ mu esse so se</td>
</tr>
<tr>
<td></td>
<td>&lt;SOS&gt; this field was near a pond which was full of frogs &lt;EOS&gt;</td>
<td>saabɛnatɛkyɛ bi a ahɛyɛhɛyɛhɛyɛ mu mu &lt;EOS&gt;</td>
<td>saasareyibenɛatɛkyɛ bi a mpoʃɔraɛhɛyɛ mu ma ho</td>
</tr>
<tr>
<td></td>
<td>&lt;SOS&gt; i am buying something &lt;EOS&gt;</td>
<td>me reɔrɛtɛ &lt;EOS&gt;</td>
<td>me reɔrɛdɛ &lt;EOS&gt;</td>
</tr>
</tbody>
</table>

Commission of Shanghai Municipality under Grant 17YF1427400, the Fundamental Research Funds for the Central Universities under Grant 17D111206, the Research on Social Sciences Development in Hebei Province under Grant 20200302075 and the Marine Science Research Project of Hebei Normal University of Science and Technology under Grant 2018HY020 for their support.

REFERENCES


Sun H, Jafar SA (2017). The capacity of private information retrieval, IEEE