LLMs with Low-Resource Translation: Syriac-to-English Case Study

Stanford CS224N Custom Project

Andrew T. Lee
Department of Computer Science
Stanford University
adtlee@stanford.edu

March 17, 2024

Abstract

Translation models perform well when working with highly-digitized languages, such as English, Mandarin, and German. However, not as much work has been done to evaluate their performances on low-resource languages. To add on to this literature of low-resource translation, this project investigates existing translation models' performances on Syriac to English (sy→en). Experimenting with LSTM NMT, Transformer NMT, fine-tuned T5, fine-tuned GPT, and LSTM NMT with back-translation, I find that LSTM NMT performs the best amongst the 5 models on this task, achieving the highest METEOR and BERT-Score. Qualitatively, the LSTM NMT model's generations also appear the most semantically accurate.

1 Key Information

- Mentor: Soumya Chatterjee

2 Introduction

Within the past few years, many breakthroughs have taken place in the field of natural language processing (NLP). With the development of BERT, ChatGPT, and other large language models (LLMs), we see growing success with bigger models and larger datasets. Such models have become quite capable in tasks like translation, content generation, question answering, summarization, and sentiment analysis. However, not as much research has been conducted to evaluate LLMs' performances in low-resource contexts, where they have a much smaller dataset to work with. Adding on to the field of low-resource NLP, this project presents a case study of low-resource machine translation. Using prominent LLM architectures, I constructed 5 models and evaluate their performance on the task of Syriac to English (sy→en) translation.

To motivate the study of sy→en translation, allow me to share some background information on Syriac. The history of pre-modern Christianity often concentrates on Western Christianity, the predecessors of Roman Catholicism and Protestantism. However, in the early and medieval periods, the most geographically expansive churches were actually Syriac Christian communities, extending from Eastern Turkey, throughout the Middle East, through Afghanistan, Tibet, into China and down into India (Penn et al., 2022). These Christians spoke and wrote in Syriac (ܠܫܢܐ ܣܘܪܝܝܐ), an ancient dialect of Aramaic. Knowledge of Syriac is thus essential for understanding the culture and history of pre-modern Christianity.

Unfortunately, few pedagogical resources for learning Syriac exist today, and this language barrier leads many historical texts to be inaccessible to both scholars and communities with Syriac lineage. Therefore, in addition to being a NLP experiment, this project hopes to contribute to an ongoing effort to preserve Syriac by providing sy→en translation tools.

3 Related Work

3.1 Machine Translation

Machine translation is the task of translating a sequence of text from a source language to a target language through LMs. Cho et al. (2014) and Sutskever et al. (2014) introduced the use of a recurrent neural network (RNN) architecture for machine translation, leading to the rise of Neural Machine Translation (NMT) systems. Wu et al. (2016) then suggested the use of Long Short-Term Memory (LSTM) networks to capture more long-range dependencies between words, enabling the translation of longer sequences. Later on, Vaswani et al. (2017) introduced the Transformer architecture, revolutionizing the NLP field with its attention mechanism, which excels at long-range dependencies and complex word relationships. With the success of Transformers, models like BART by Lewis et al. (2019) and T5 by Raffel et al. (2023) have been developed, allowing for multiple NLP tasks to be completed by a single model.

3.2 Low-Resource Translation

One limitation faced by many of these translation models is the reliance on large parallel datasets between the source and target language. However, such corpora is often unavailable for many language pairings. One solution is Unsupervised Machine Translation (UMT), proposed by Lample et al. (2018). This method uses monolingual data from both languages to create a shared latent space. Sentences from either languages can then be mapped into embeddings, which are then converted into the other language. Similarly, Faheem et al. (2024) proposed semi-supervised MT, a hybrid technique combining UMT with the regular supervised procedure using a parallel dataset. Another common approach to low-resource translation is back-translation (BT), proposed by Sennrich et al. (2015). Under this framework, a back-translator model translates monolingual data of the target language into synthetic data in the source language. This additional data can then be used to provide more parallel data for training the actual translation model.

3.3 Syriac Translation

In the specific domain of Syriac translation, Augin Aydin et al. (2023) developed the first AI-powered Syriac translator, Syriac.IO. Their model allows for multilingual and bi-directional translation, such as Syriac to German or French to Syriac. However, upon personal evaluation, their model does not translate Syriac texts very well in terms of maintaining semantic accuracy. Additionally, their interface has a 150-character limit in both source and target sequence length, which poses as a considerable constraint for translating slightly longer sentences. The team has not released any information on the model or dataset used for their translator, so I attempted to contacting their developers. Unfortunately, they did not respond to me. Nonetheless, their model is accessible online, so I used it as the baseline for this study.

4 Approach

For this study, I constructed 5 sy→en models: LSTM NMT, Transformer NMT, fine-tuned T5, fine-tuned GPT, and LSTM NMT with back-translation.

4.1 Simple NMTs

The first model, **LSTM NMT**, is a sequence-to-sequence (Seq2Seq) network with attention, using a bidirectional LSTM Encoder and a Unidirectional LSTM Decoder. As the starting point of my project, this model's implementation was taken directly from my submission for Assignment 4.

With the rising popularity of Transformers in NLP work, the second model, **Transformer NMT**, employs a Transformer encoder-decoder, following the architecture proposed by Vaswani et al. (2017). To construct this model, I modified the code for LSTM NMT, replacing the LSTM encoder and decoder layers with PyTorch's `nn.Transformer` module. The projection layers were removed, and a new positional encoder was added by following PyTorch's World Language Model example implementation (PyTorch, 2023). An additional linear layer was added to convert the decoder outputs into the correct format as well. I also modified the functions in the Transformer NMT class to adjust to this new architecture.

For both models, I used the same vocabulary generated by a SentencePiece tokenizer. Both the source and target language vocabularies have 8000 subwords, a fitting size for a dataset with only around 35,000 examples.

4.2 Fine-tuned Models

For the next two models, I explored the path of fine-tuning, where I trained pre-trained models on sy$\rightarrow$en translation. While these models have not been exposed to Syriac texts before, their previous exposure to English and translation tasks could be beneficial. **Fine-tuned T5** used huggingface's pre-trained `google-t5/t5-small` model. Developed by **?**, this model was pre-trained on NLP tasks related to English, French, Romanian, and German. I used a T5 Tokenizer, based on SentencePiece, to generate its vocabulary. For its implementation, I employed huggingface's `datasets`, `transformers`, `evaluate` libraries to prepare the data and set up the model's parameters.

Within the past two years, ChatGPT has significantly risen in its popularity, becoming the new state-of-the-art LLM. **Fine-tuned GPT** thus used ChatGPT as its foundation. Using OpenAI's fine-tuning feature, I set up a new fine-tuning task using gpt-3.5-turbo. The training and validation datasets were sent to the OpenAI server to fine-tune the model. Using their API, I then called the fine-tuned model to translate the testing dataset.

4.3 Back-translation NMT

The last model, **LSTM NMT with BT**, explores the method of back-translation. Although there is not much parallel sy$\leftrightarrow$en data, there is an abundance of monolingual English texts. Synthetic parallel data could thus be generated to create a larger training dataset for the sy$\leftrightarrow$en model. I then trained a BT model that performs en$\rightarrow$sy translations. I used it to translate a monolingual English dataset into synthetic Syriac texts. The resulting dataset, comprised of synthetic Syriac sentences and actual English sentences, was then added to the existing parallel training corpus. LSTM NMT with BT then used the expanded corpus as training data.

5 Experiments

5.1 Data

For this study, I used the Bible for my dataset, as it is the only available parallel corpus for sy$\rightarrow$en. However, since Syriac is a liturgical language, this should mitigate some issues related to narrow datasets. The Syriac version of the Bible is called the Peshitta, which is made available on Github by Roorda and van Peursen (2021). For its English counterpart, I used the King James Version Bible (KJV) (SuperSearch, 2006). Although KJV is not a direct translation of the Peshitta, the underlying meaning of the texts are mostly equivalent. In total, this corpus contained 34,800 paired verses. 2,000 of those verses were taken out for validation and evaluation, so the training corpus contained 32,800 sy$\rightarrow$en examples: $Bible_{\text{train}} = (Bible_{\text{train-sy}}, Bible_{\text{train-en}})$. This served as the training data for LSTM NMT and Transformer NMT.

For Fine-tuned T5 and Fine-tuned GPT, smaller training and validation sets were used due to computational constraints. I trained Fine-tuned T5 with $100 Bible_{\text{train}}$ and $100 Bible_{\text{dev}}$, each file containing 100 examples. As for Fine-tuned GPT, I used $200Bible_{\text{train}}$ and $200Bible_{\text{dev}}$, each file now containing 200 examples. Additionally, since these models were pre-trained to do a variety of NLP tasks, I added a prefix "Please translate this Syriac text to English" to each of the Syriac inputs.

For LSTM NMT with BT, I trained the back-translator with $Bible_{\text{train}}$. I then asked the back-translator to translate the Corpus of Early English Correspondence Sampler (CEECS), which contained 41102 lines of text from letters written in 1418-1680 (cee, 2003). I originally wanted to use a corpus of English religious texts, but such corpora were not free to use. Nonetheless, the texts in CEECS should suffice since they are written in the same time period as KJV. Amongst the Syriac translations, I removed ones that have the `<unk>` token constitute more than 10% of the tokens in the sequence. This left a remainder of 33,173 examples in $CEECS = \{CEECS_{\text{sy}}, CEECS_{\text{en}}\}$. The training dataset for LSTM NMT with BT combined $Bible_{\text{train}}$ with $CEECS$, giving an expanded corpus $Expanded = \{Bible_{\text{train-sy}} + CEECS_{\text{sy}}, Bible_{\text{train-en}} + CEECS_{\text{en}}\}$ of 65,973 examples.

The same testing dataset $Bible_{\text{test}}$ was used for all 5 models.

5.2 Evaluation method

A standard evaluation metric for machine translation, I used BLEU, which emphasizes word accuracy (Papineni et al., 2002). I also used METEOR, which takes into consideration account fluency and word order, features not captured by BLEU. (Banerjee and Lavie, 2005).

Lastly, I used BERT-Score by Zhang et al. (2020), which focuses more on the underlying semantics between model translations and the reference translations. This is important because the baseline, Syriac.IO, was not trained on the same dataset as my models, so its translations would not necessarily follow the wording in KJV, even if the translations are semantically correct. BLEU and METEOR might give the baseline lower scores as a result. For the evaluation model, I used "roberta-large," with the scores rescaled to fit in a $[-1, 1]$ range and inverse document frequency not applied.

5.3 Experimental details

For LSTM NMT, I used an embedding size of 1024, a hidden size of 768, a batch size of 32, and a dropout rate of 0.2. For training, I used an Adam optimizer and set the learning rate to 0.00001, patience to 5, `valid-niter` to 100, and `max-epoch` to 1,000. For Transformer NMT, I used embedding and hidden sizes of 256, a batch size of 32, and a dropout rate of 0.2. I set up the Transformer with 2 multi-heads, 3 encoder layers, and 3 decoder layers. The training configuration was the same as LSTM NMT.

For Fine-tuned T5, since the model used a much smaller dataset, I changed the training parameters to a learning rate of 0.0001, a batch size of 4, a dropout rate of 0.1, and a `max-epoch` of 100. For Fine-tuned GPT, no paramter control is provided to users on the API, so I left the configuration as default.

For both BT and LSTM NMT with BT, the model and training configuration were the same as those of LSTM NMT. The BT itself followed the LSTM NMT architecture, which outperformed one using a Transformer architecture, as documented in Table 1. Additionally, in the study by Sennrich et al. (2015), no parameters were frozen during training. In this case, however, the synthetic Syriac sentences were incredibly noisy, which could end up hindering the model's learning, causing it to un-learn the good parameterizations. Therefore, I tried creating two LSTM NMT with BT models: (a) freezing the encoder parameters when given data from the synthetic dataset, and (b) treating all data equally. In Table 2, we see that method (a) resulted in better overall performance, so I proceeded with that model instead.

BT Model	BLEU
LSTM NMT	12.8103%
Transformer NMT	7.2385%

Table 1: BLEU scores of back-translation models.

LSTM NMT with BT Model	BLEU	METEOR	F_{BERT}
freeze encoder parameters	10.06889%	42.3203%	32.3651%
unfreeze encoder parameters	11.2485%	40.4795%	29.5961%

Table 2: BLEU, METEOR, and F_{BERT} scores of LSTM NMT with BT models.

5.4 Results

The BLEU and METEOR scores for the baseline and my models are listed in Table 3.

Model	BLEU	METEOR
Syriac.IO [baseline]	**13.2592%**	12.3681%
LSTM NMT	11.2826%	**43.4169%**
Transformer NMT	10.4898%	11.6030%
Fine-tuned T5 (100Bible)	9.9994%	12.1287%
Fine-tuned GPT (200Bible)	10.5941%	14.8462%
LSTM NMT with BT	11.2485%	42.3203%

Table 3: BLEU and METEOR scores of sy→en models.

Unexpectedly, no model was able to beat the baseline in terms of BLEU score. This is rather ironic. Earlier, I stated that BLEU could be an unfair metric to Syriac.IO, since it was not trained on the same dataset as my models, which is in the same corpus as the testing set, so the baseline's translations may be still correct but worded differently. Yet, it achieved the highest BLEU score across all models. Perhaps Syriac.IO trained on a dataset with English texts written in a similar time period to KJV, allowing it to produce English translations that are stylistically similar to those in KJV. Comparing my models, we see that the range of scores is rather narrow, from 9.9994% to 11.2826%, suggesting that all models generate about the same amount of "correct" words. These scores are also fairly low, indicating that they all struggled to deliver syntatically accurate translations. This overarching behavior was most likely due to the lack of a large dataset.

When comparing METEOR scores, however, we see a significant increase in LSTM NMT and LSTM NMT with BT's performance, outperforming all other models by nearly 30%. Since METEOR places more emphasis on word ordering, this suggests that the LSTM architecture produced texts that were more grammatically structured, generating phrases that better align with the gold-standard translations. In contrast, the other models, including the baseline, performed rather poorly here. Upon inspection of the actual translations, which I will go into more detail in the Analysis section, I hypothesize that their low scores were due to heavily repetitive and syntactically different translations.

Model	P_{BERT}	R_{BERT}	F_{BERT}
Syriac.IO [baseline]	2.6785%	2.2642%	0.2995%
LSTM NMT	**37.1195%**	**33.3298%**	**35.2575%**
Transformer NMT	-15.3805%	-10.5349%	-12.9885%
Fine-tuned T5 (100Bible)	-44.1273%	-12.6694%	-29.1455%
Fine-tuned GPT (200Bible)	0.6905%	1.5630%	1.1943%
LSTM NMT with BT	34.5652 %	30.1021%	32.3651%

Table 4: BERT-Scores (Precision, Recall, F1) of sy→en models.

A similar situation appears for BERT-Scores, documented in Table 4. This was quite shocking to me. BERT-Score captures semantic similarity between candidate and reference translations, so it should score a semantically accurate translation highly even if its wording is different. However, the baseline performed rather poorly here, retrieving a F_{BERT} score of nearly 0%. This suggests that its translations did not semantically align with the reference translations. Fine-tuned GPT gave a similar result, delivering coherent, but semantically inaccurate translations. The two Transformer models performed even more terribly, obtaining negative BERT-Scores. This makes sense though, since—as we will see next—their translations were nonsensical and extremely repetitive.

Overall, these scores suggest that the LSTM models performed the best in terms of maintaining grammatical structure and sentence meaning. One possible explanation for them to outperform Transformers might be the size of the dataset. Considering Transformers' internal architecture, the relationships between inputs and outputs are more complex that those in a LSTM encoder-decoder, so it might take a much larger dataset for a Transformer to train effectively.

The fine-tuned T5 also did not seem to benefit from its pre-training at all, since it behaved the same as a Transformer trained from scratch. This was likely due to its usage of a new vocabulary. Since its vocabulary had changed, the skills it picked up from pre-training were essentially lost too, making it no different from a untrained model. Additionally, it used a much smaller dataset than other models, having access to only 100 training examples. Similarly, the fine-tuned GPT model only had access to 200 training examples, so it makes sense that its performance was mediocre. Even so, when examining its scores across the three categories, it is on-par with the baseline, perhaps even slightly better. If the full dataset $\textit{Bible}_{\text{train}}$ was given to this model, perhaps it would have been the best model amongst the five.

Lastly, while LSTM NMT with BT performed relatively well, it was consistently worse than LSTM NMT. This indicates that back-translation was ineffective in this study, perhaps even making the model perform worse. The synthetic data was probably too noisy. After all, the back-translator only had a BLEU score of 12.8103%, as shown in Table 1. Since half of *Expanded*'s examples use synthetic source sentences made by this back-translator, it was more likely that those examples confused LSTM NMT with BT than helped it learn.

6 Analysis

I selected five testing examples that best capture the overall behavior of the models, documented in Tables 5, 6, 7, 8, and 9 (Tables 7-9 are in the Appendix). Amongst these examples, LSTM NMT and LSTM NMT with BT delivered translations that best correlate with the ground truth references.

6.1 LSTM NMT and LSTM NMT with BT

Overall, LSTM NMT and LSTM NMT with BT provided translations that semantically match the ground truth the most, supporting the results from METEOR and BERT-Score. The translations of both were often quite similar, but LSTM NMT tended to provide slightly better translations than LSTM NMT with BT, such as in Table 5, where the BT-based model replaced "strength" with "help," giving a less accurate translation. This reinforces BT's worsening effect on LSTM NMT, though the influence was trivial. Grammatically, these two models' translations matched the ground truth fairly well, and they were also quite coherent. However, both struggled to grasp longer-range dependencies, such as how "our help" and "our shield" should refer to "he" in Table 5.

6.2 Transformer NMT and Fine-tuned T5

Transformer NMT and Fine-tuned T5 performed the worst across all examples. Examining Transformer NMT's translations, we see that it was only generating sentences from a small collection of words and phrases ("children", "the lord", "and", etc.). Transformer NMT likely gave these few tokens extremely high probabilities, such that the Syriac inputs hold little influence in determining what the English translation would be. Its translations were therefore nonsensical and heavily repetitive. This behavior was also present in Fine-tuned T5, excessively using phrases like "he said" and "unto the lord". Interestingly, Fine-tuned T5 occasionally generated some Syriac tokens in its translations too, as seen in Table 6. This was likely due to the nature of T5's tokenizer, which shared its vocabulary between source and target sentences. In this context though, this approach led it to perform the worst, since it could not even fully discern English words from Syriac words.

6.3 Fine-tuned GPT

Fine-tuned GPT produced very coherent translations, reflecting the syntatical effectiveness of using a pre-trained model. However, its translations deviated heavily from the ground truths when it came to semantics. GPT learned to generate biblical-sounding English texts, but these generations did not relate to the source sentences at all. This behavior was apparent across all examples.

Model	Text
Source	ܢܦܫܢ ܣܟܝܐ ܠܡܪܝܐ ܡܛܠ ܕܗܘܝܘ ܡܥܕܪܢ ܘܡܣܝܥܢ
Ground Truth (KJV)	our soul waiteth for the lord: he is our help and our shield.
Syriac.IO [baseline]	i dont want to lie for the sake of the lord for you
LSTM NMT	we wait for the lord: for he is our strength, and we are our help.
Transformer NMT	and the children of the children of the lord.
Fine-tuned T5 (100Bible)	then he with be unto the lord, and shall be and shall women:, and shall›
Fine-tuned GPT (200Bible)	O lord, thou hast searched me, and known me.
LSTM NMT with BT	we wait for the lord: for he is our help, and we are our help.

Table 5: Translation comparisons 1 for sy→en models.

Model	Text
Source	ܘܬܡܝܗܝܢ ܗܘܘ ܟܠܗܘܢ ܐܝܠܝܢ ܕܫܡܥܝܢ ܗܘܘ ܠܗ ܒܚܟܡܬܗ ܘܒܦܬܓܡܘܗܝ
Ground Truth (KJV)	and all that heard him were astonished at his understanding and answers.
Syriac.IO [baseline]	who was a godly when they heard him
LSTM NMT	and all they that heard him were all amazed, who heard him in his own sight.
Transformer NMT	and he, and the sons of the children of the earth.
Fine-tuned T5 (100Bible)	and he said, that ܗܘܘ ܐܝܠ him that we ܠܗ
Fine-tuned GPT (200Bible)	and all people of the earth shall see that thou art called by the name of the lord; and they shall be afraid of thee.
LSTM NMT with BT	and they were all amazed by him in the way of the lord.

Table 6: Translation comparisons 2 for sy→en models.

7 Conclusion

In this work, I compared the performances of LSTM NMT, Transformer NMT, fine-tuned T5, fine-tuned GPT, and LSTM NMT with BT on low-resourced sy→en translation. My results show that LSTM NMT delivered the most accurate translations, both in terms of semantics and METEOR and BERT-Scores. However, the baseline model achieved the highest BLEU score.

Transformer-based models performed poorly in this study, with major repetition issues. Future studies can explore using more diverse sy→en datasets, which might significantly change these models' performances. Additionally, since Fine-tuned T5 used a significantly smaller dataset than other models, future work should try using a larger dataset for fairer comparison. Fine-tuned T5 also used a new vocabulary, making its pre-trained skills useless. Rather than building a new vocabulary entirely, it may be worthwhile to explore adding new source and target words to an existing vobaulary instead.

Fine-tuned GPT delivered coherent translations, but they did not align with the semantics of the reference sentences. This was also likely due to the use of a smaller dataset. Future work can explore fine-tuning GPT with a much larger corpus. Lastly, BT was ineffective in this study, even making NMT perform worse at times. This could be due to an unreliable en→sy BT and a bad monolingual target corpus. Future research should investigate using different BTs and monolingual corpora that better align with the content in the parallel corpus.

References

2003. Corpus of early english correspondence sampler (CEECS). Oxford Text Archive.

Polycarpus Augin Aydin, Jan Bet-Sawoce, John Kaninya, Shabo Talay, Sebastian Kenoro Kiraz, Mur U La Mur, Yakup Cekici, Benjamin Yanik, Samuel Balci, Saroy, Christopher Mrani, and Gabriel Mrani. 2023. Syriac.io. Online.

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan. Association for Computational Linguistics.

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation.

Mohamed Atta Faheem, Khaled Tawfik Wassif, Hanaa Bayomi, and Sherif Mahdy Abdou. 2024. Improving neural machine translation for low resource languages through non-parallel corpora: a case study of egyptian dialect to modern standard arabic translation. *Scientific Reports*, 14(1):2265.

Guillaume Lample, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2018. Unsupervised machine translation using monolingual corpora only. In *International Conference on Learning Representations*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. pages 311–318.

Michael P. Penn, Scott F. Johnson, Christine Shepardson, and Charles M. Stang. 2022. *Invitation to Syriac Christianity : an anthology*. University of California Pres.

PyTorch. 2023. Word-level language modeling using rnn and transformer.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2023. Exploring the limits of transfer learning with a unified text-to-text transformer.

Dirk Roorda and Wido van Peursen. 2021. Peshitta. In *Peshitta*. Eep Talstra Centre for Bible and Computer.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Improving neural machine translation models with monolingual data.

Bible SuperSearch. 2006. The bible (king james version). In *The Bible*.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. Neural Information Processing Systems.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Łukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

A Appendix

Model	Text
Source	ܛܘܒܝܟܘܢ ܐܡܬܝ ܕܡܚܣܕܝܢ ܠܟܘܢ ܘܪܕܦܝܢ ܠܟܘܢ ܘܐܡܪܝܢ ܥܠܝܟܘܢ ܟܠ ܡܠܐ ܒܝܫܐ ܡܛܠܬܝ ܒܕܓܠܘܬܐ
Ground Truth (KJV)	‹blessed are ye, when› men ‹shall revile you, and persecute› you, ‹and shall say all manner of evil against you falsely, for my sake.›
Syriac.IO [baseline]	blessed are the children of knowing that they might be saved and say we have crucified you out of war according to the weakness of his death
LSTM NMT	happy are ye, when they say unto you, persecute you, and persecute you, and say unto you, every one that is evil of his own sake, and for my sake; for he hath sinned against him.
Transformer NMT	and he said unto him, and said, ‹and thou shalt thou shalt be in the lord.
Fine-tuned T5 (100Bible)	and he said said, that he with that he shall cast with ܡܠܟܐ, and they had out of the land of the land of the land of the land of the earth, and they had shall cast with him.
Fine-tuned GPT (200Bible)	behold, i descend unto you in judgment; and i will be a swift witness against the sorcerers, and against the adulterers, and ..., and that turn aside the stranger from his right, and fear not me, saith the lord of hosts. lord; and they shall be afraid of thee.
LSTM NMT with BT	‹blessed› are ‹blessed› men, ‹when it shall be unto you, and they will speak unto you, saying, all that evil is evil; for my soul is full of vain.

Table 7: Translation comparisons 3 for sy→en models.

Model	Text
Source	ܐܡܪܝܢ ܠܗ ܝܫܘܥ ܢܨܪܝܐ ܥܒܪ
Ground Truth (KJV)	and they told him, that jesus of nazareth passeth by.
Syriac.IO [baseline]	they said jesus of nazareth will be scattered
LSTM NMT	and they said unto him, jesus of nazareth.
Transformer NMT	and the lord,
Fine-tuned T5 (100Bible)	and the after of the lord said, and his did said the lord of the land of the earth, and his did said the lord of the lord
Fine-tuned GPT (200Bible)	then said jesus unto him, except ye see signs and wonders, ye will not believe.
LSTM NMT with BT	and they said unto him, jesus of nazareth,

Table 8: Translation comparisons 4 for sy→en models.

Model	Text
Source	ܘܫܡܥܘ ܬܪܝܗܘܢ ܬܠܡܝܕܘܗܝ ܟܕ ܐܡܪ ܘܐܙܠܘ ܠܗܘܢ ܒܬܪܗ ܕܝܫܘܥ
Ground Truth (KJV)	and the two disciples heard him speak, and they followed jesus.
Syriac.IO [baseline]	according to the works of the same flesh he says i will join you to jesus
LSTM NMT	and when the disciples heard it, he spake unto them, and followed him.
Transformer NMT	and the lord, saying, the lord,
Fine-tuned T5 (100Bible)	and the lord of the lord of the lord of the lord of the then and his times he did the lord of the lord of the lord of the lord of the lord
Fine-tuned GPT (200Bible)	if they obey and serve him, they shall spend their days in prosperity, and their years in pleasures.
LSTM NMT with BT	and when his disciples heard it, he said unto them, after jesus was the son of jesus.

Table 9: Translation comparisons 5 for sy→en models.