Pivoting in Low Resource Environments

Adytia Vetukuri (g01213246), Naveen Porla (g01221341), Matthew Burnard (g01099938)
avetukur@gmu.edu, nporla@gmu.edu, mburnard@masonlive.gmu.edu

Abstract

In attempting to solve the low resource problem between Azerbaijani and English by first pivoting to Turkish we found that directly translating through both steps produces a very strong accumulated error effect. We also found that using back translation from a related language lessens or sidesteps that issue, resulting in modest increases in Bleu score.

Introduction

In modern machine translation recurrent neural networks reign supreme. This broad category of machine learning algorithms exhibit some special characteristics that make it an excellent choice for machine translation: they can accept variable length input and they can take into account previous network states in their predictions. Recently, the LSTM has shown to be especially adept at machine translation, and specifically the sequence to sequence (seq2seq)[1] style of encoder-decoder translation. But this class of neural networks, while able to approximate an astonishing level of problem complexity, has 1 major downside: it requires a lot of high quality data to train on[2].

While this requirement is attainable for some languages pairs with large speaking populations, computer science educated speaking populations, and/or well funded speaking populations it remains a huge barrier for low population, low education opportunity, and/or poorly funded language groups. As such, the data available to train models for these groups in relation to an unrelated target language is either low in quantity, low in quality, or frequently both. This severely limits the applicability of seq2seq machine translation models.

One potential approach, laid out in *Generalized Data Augmentation for Low-Resource Translation*[3] shows promise. Instead of training a model in the low resource environment between source and target you train 2 models: one to translate from the source language to a pivot language, and another to translate from the pivot language to our target language. What sort of advantage would this offer? Well, if the pivot language has a high resource environment with the target language *and* some combination of high resource and/or linguistic similarity with the source language then it may overcome the accumulated error from multiple translation steps to produce a higher quality translation than just source to target.

---

[1] Singh, *Neural Machine Translation using Simple Seq2Seq Model*. 2020.

[2] Sutskever et al., *Sequence to Sequence Learning with Neural Networks*. 2014

[3] Xia et al., *Generalized Data Augmentation for Low-Resource Translation. 2019.*

Previously

In our previous paper we discussed using relatively simple, trained from scratch encode-decoder models. We found this approach to be insufficient to overcome domain issues associated with our primary source of data and the accumulated error inherent with pivoting between multiple languages.

Datasets

For datasets we used 3 parallel corpora of varying size in Azarbaijani-English (az-en), Azarbaijani-Turkish (az-tr), and Turkish-English (tr-en) language pairs.

Tanzil - The Tanzil dataset is built on the Quran; the Islamic Holy Book. Since the book has been translated into every major language, and a vast number of minor languages, it presents an opportunity to collect a large quantity of parallel data between many languages. The primary issue with this, and indeed all holy text datasets, is that the language used in it is frequently only loosely representative of the languages in question.

http://opus.nlpl.eu/Tanzil-v1.php

Tatoeba - A small dataset consisting of short sentences translated by volunteers between many languages.

http://opus.nlpl.eu/Tatoeba-v2020-05-31.php

Open Subtitles 2018 (tr-en only) - A massive parallel database of subtitles for TV shows and movies. It presents a good source of translation, but the number of languages available is limited.

http://opus.nlpl.eu/OpenSubtitles-v2018.php

Experiments

We focused on 3 main topics for further experimentation. Teacher forcing, Transformer models, and fine-tuning pretrained models.

Teacher Forcing:

We noticed that our models are predicting repeating words during the initial phases of training. This caused the recurrent neural networks to converge slowly and resulted in poor performance, as these sequence prediction models use the output from previous time step t-1 as an input for the current time step t. Due to this the hidden states of the model were updated by a sequence of wrong predictions thus accumulating errors.

To handle this we employed the teacher forcing strategy. Teacher forcing is a technique to pass the actual ground truth to the model and use this ground truth as an input at time step t instead of using the output generated by the network. This helps the model to learn quicker and makes the training process converge faster.[4] Although training with teacher forcing is good strategy in training sequence prediction models, using the ground truth at each time step during training the model will not be able to learn how to handle repetition of words[5] which does occur in our datasets. So we used a threshold of 0.4 as our teacher forcing ratio where we pass the actual ground truth 40% of times and network predicted output 60% of times.

Seq2seq with teacher forcing model parameters: Embedding dimensions = 200, Hidden dimensions = 512, Adam optimizer(lr=0.001), teacher forcing ratio = 0.4, Batch size = 32(sentences), source/target vocab = 20000.

Model	**Training / Test dataset size**	**Epochs / minutes per epoch**	**Embedding / Hidden Dimensions**	**Model Name/Output filename**	**Bleu Score**
AZ-EN	100k/ 10k	50/14 minutes	200/512	seq_az_en_100k.pth.tar	16.143
AZ-TR	200k/ 20k	33/15minutes	200/512	seq_az_tr_200k.pth.tar	11.65
TR-EN	200k/ 20k	40/14minutes	200/512	seq_tr_en_200k.pth.tar	16.55
AZ-EN (Direct Translation)	**20k Testset**	**-**	**-**	**seq_az_en_direct_eng20k.txt (Direct translation)**	**11.06**
AZ -> TR -> EN (Pivot Translation)	**20k Testset**	**-**	**-**	**seq_az_tr_pivot_turk20k.txt/ seq_tr_en_pivot_eng20k.txt (Pivot translation)**	**9.29**

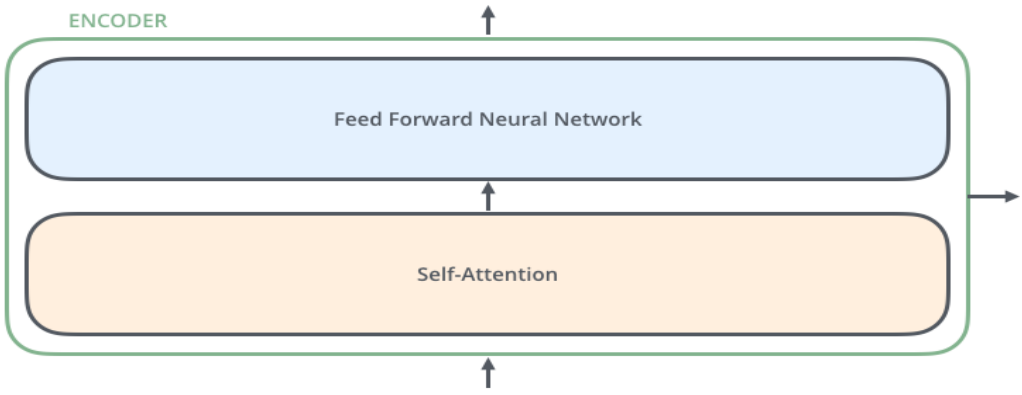
[4] Wanshun Wong, *What is Teacher Forcing?*. 2019.

[5] Jason Brownlee, *What is Teacher Forcing for Recurrent Neural Networks*. 2019.

Pytorch Transformer Model

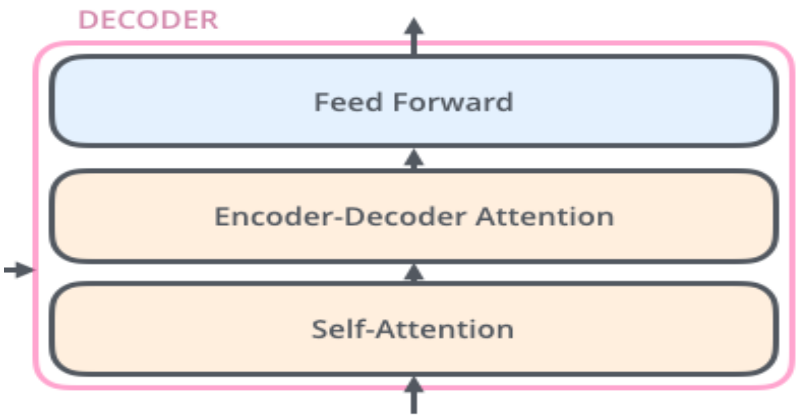
The transformer model proved to be an excellent tool for machine translation, especially given how important attention is to the task. We tested the pytorch standard transformer model.[6] To solve the issue of parallelization transformers use convolutional neural networks in conjunction with self attention. (All illustrations from[7])

Internally our transformer has a similar architecture as our previous seq2seq model but now we have 3layers of encoders and 3 layers of decoders. Each encoder is very similar to each other and they consist of two layers of self-attention and a feed forward nerural network. The self attention layer is important to help the encoder look at other words in the input sequence. The decoder has both the above layers, but with an additional attention layer between to help the decoder focus on the relevant information.



[6] PyTorch Documentation, *Sequence-to-Sequence Modeling with NN.Transformer and TorchText*. 2017.

[7] Jay Alammar, *The Illustrated Transformer*. 2018.



The full model consists of three encoder and three decoder layers, each with 8 attention heads. In comparison to the previous seq2seq model, which took as long as 13 hours to train and failed to produce a usable model, the transformer model both trains faster and achieves higher bleu scores. This is in part because the seq2seq model is insufficient for languages with large vocabularies like Azerbaijani and Turkish.

The Az-En direct translation model, trained on 100k sentences, as able to achieve a bleu score of 15.95 on the 20k sentence test set. For the pivot tests we independently trained Az-Tr and Tr-En models with 200k sentences, and passed the same 20k sentence test set through the pivot. This produced bleu scores of around 12.9, a slight decrease over the direct model. This suggest each model is roughly equivalent and that the accumulated error during the pivot step is still an issue.

Xia et al. found using data back translated from a related language to train a direct source-target model produces improvements over a base model of +1 to +2 Bleu score.[3] When 200k additional Tanzil sentences are translated into Azerbaijani from Turkish, using a model trained for the task, and added to the initial training set of 200k sentences from Tanzil we experienced an increase of around +1 Bleu score over the original direct model.

Model Parameters: encoder layers = 3, decoder layers = 3, num heads = 8, dropout = 0.1, embedding size = 200, feed forward dimensions = 512, Adam optimizer(lr=0.0005), Batch size = 32 (sentences).[8]

Results from Literature: Azerbaijani to English using Turkish as pivot

Standard NMT	11.83
Standard supervised back-translation	12.46
Data Augmentation	**15.91**

Results from Transformer Models:

Model	**Training/ Test size**	**Epochs / minutes per epoch**	**Model Name/ Translated Output file name**	**Bleu Score**
AZ-EN	100k/10k	50/3 minutes	tf_az_en_100k.pth.tar	18.04
AZ-TR	200k/10k	60/4 minutes	tf_az_tr_200k.pth.tar	15.33
TR-EN	200k/10k	60/5 minutes	tf_tr_en_200k.pth.tar	23.91
TR-AZ (Back-translation model)	200k/10k	60/4minutes	tf_tr_az_200k.pth.tar	32.63
AZ-TR (using 200k back-translated data)	400k/10k	50/12 minutes	tf_az_tr_400k.pth.tar	14.94
TR-EN (No back translation required as tr-en has many resources)	400k/10k	42/10 minutes	tf_tr_en_400k.pth.tar	21.70
AZ-EN (Direct Translation)	**100k/ 20k Testset**	**-**	**tf_az_en_direct_eng20k.txt**	**15.95**
AZ -> TR -> EN (Pivot Translation)	200k/ 20k Testset	-	tf_az_tr200k_pivot_turk20k.txt / tf_tr_en200k_pivot_eng20k.txt	12.89
AZ -> EN (Pivot Translation) Back translated data	**400k/ 20k Testset**	**-**	**tf_az_tr400k_pivot_turk20k.txt/ tf_tr_en400k_pivot_eng20k.txt**	**16.69**

[8] Aladdin Persson, *Machine-Learning-Collection/.../seq2seq_transformer.py* 2020.

Fine-tuning pretrained Models with related language back-translated data

Huggingface is a popular toolkit for many NLP tasks, and the MarianMT pretrained models at their base perform good out of the box. Their Helsinki Opus models were trained specifically on Tatoeba and seem generalize fairly well out of the box to similar domain datasets like subtitles. But of course if you're going to use a pretrained set you should seek to fine-tune that set on a dataset related to your task.

This is where the low resource problem comes in, since Azerbaijani *doesn't have* parallel datasets in, say, the field of subtitles with English. While a direct pivot to Turkish during the translation step is an attractive idea, previous testing shows the accumulated error is a major factor in multi-step translation. Previous testing has also shown that back translated data, even of questionable quality, measurably improve translation quality. As such, the Open Subtitle 2018 (En-Tr) library was used, along with MarianMT's pretrained Opus Tr-Az model, as a way to automatically generate new domain specific parallel data between Azerbaijani and English. This is functionally similar to pivoting. Ideally this would be tested on some new 3rd dataset of a related domain but lacking that a test set of backtranslated data was also generated.

Using Huggingface's automated finetune_trainer.py script, models were trained using between 1500 and 15000 back translated examples (in steps of 1500) and tested on both the back translated test set and Tatoeba (to chart specialization loss). Since both Tatoeba and Open Subtitles contain many sentences too short for Bleu 4 to work, Bleu scores 1-4 were recorded individually.

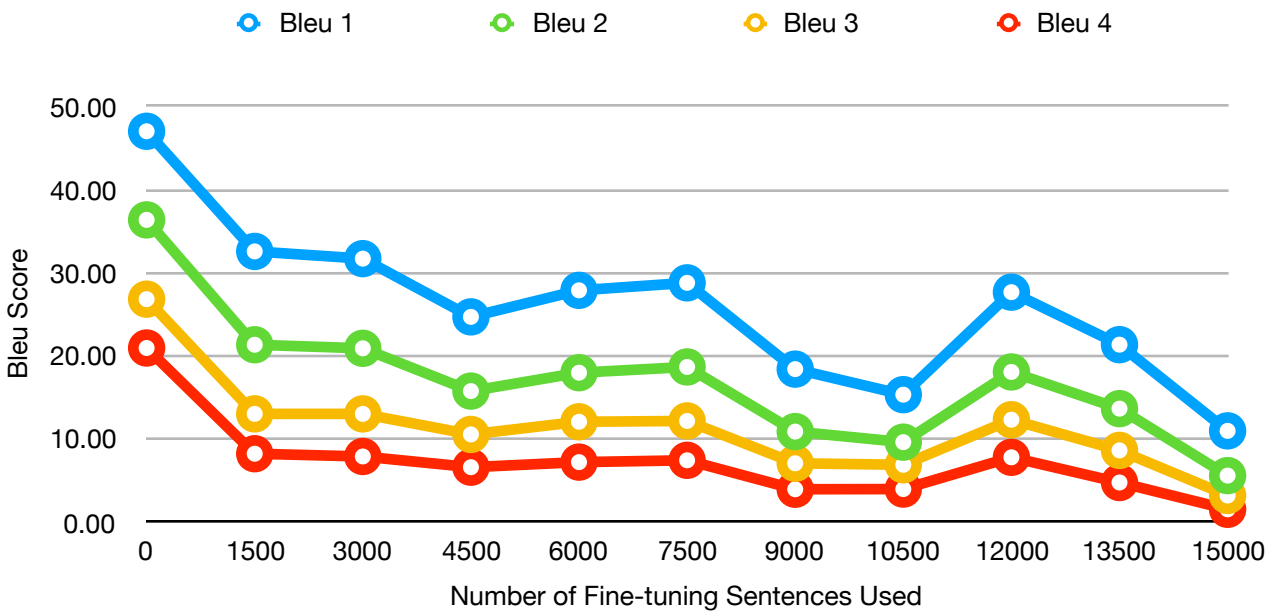
It was found that, while the initial model did relatively poorly with the back translated data, very little fine-tuning on back translated data (as little as 1500 sentences) provided +2 to +4 Bleu score on the back translated test set: a slight improvement over the +1 to +2 Bleu for the Xia et al. baseline.[3] Unsurprisingly this also lowered the accuracy on the Tatoeba test set (by a startling -12 to -15 Bleu). In fact, of all the models tried, this relatively lightly tuned model provided the greatest increase in back translated Bleu score across the board, except for the 7500 model's Bleu 3 score (which was essentially equivalent), *and* the lowest drop in Bleu score for the Tatoeba set.

Huggingface MarianMT AZ-EN Fine-tuning on Pivoted, Back-translated Data

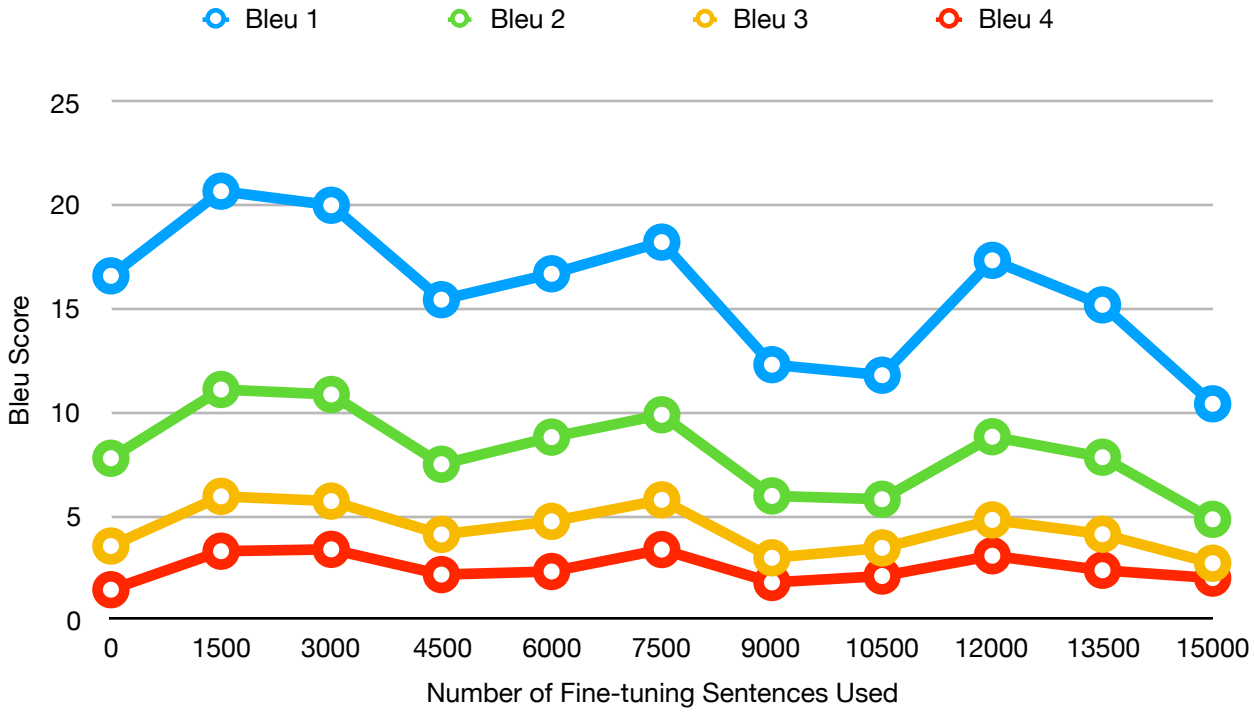
Model/Trained On	Test Set	Bleu 1	Bleu 2	Bleu 3	Bleu 4
MarianMT Base	**Tatoeba**	47.00	36.35	26.82	20.95
0	**Back Trans Test Set**	16.59	7.78	3.54	1.43
Model 1	**Tatoeba**	32.56	21.34	13.03	8.24
1500	**Back Trans Test Set**	20.68	11.10	5.94	3.29
Model 2	**Tatoeba**	31.69	20.91	13.03	7.91
3000	**Back Trans Test Set**	20.00	10.86	5.71	3.38
Model 3	**Tatoeba**	24.69	15.77	10.56	6.62
4500	**Back Trans Test Set**	15.44	7.49	4.10	2.16
Model 4	**Tatoeba**	27.88	17.98	12.08	7.23
6000	**Back Trans Test Set**	16.69	8.80	4.73	2.31
Model 5	**Tatoeba**	28.78	18.67	12.18	7.45
7500	**Back Trans Test Set**	18.22	9.88	5.77	3.37
Model 6	**Tatoeba**	18.41	10.90	7.09	3.97
9000	**Back Trans Test Set**	12.30	5.96	2.97	1.78
Model 7	**Tatoeba**	15.34	9.63	6.91	3.99
10500	**Back Trans Test Set**	11.79	5.80	3.45	2.09
Model 8	**Tatoeba**	27.66	18.07	12.33	7.81
12000	**Back Trans Test Set**	17.34	8.82	4.80	3.07
Model 9	**Tatoeba**	21.35	13.66	8.64	4.78
13500	**Back Trans Test Set**	15.19	7.83	4.11	2.37
Model 10	**Tatoeba**	10.99	5.61	3.22	1.56
15000	**Back Trans Test Set**	10.41	4.83	2.71	1.99

In an unrelated observation, in these tests, all 4 Bleu scores followed roughly the same progression, which seems to imply Bleu 1 should be sufficient for identifying plateaus, minimums, and maximums in evaluation of models. This could save on model evaluation time.

Bleu Scores of fine-tuned MarianMT models on Tatoeba Test set



Bleu Scores of fine-tuned MarianMT models on a pivot translated test set



Conclusions and Additional Observations

The biggest take-away, supported by both this and the previous paper, is that translation through a related pivot language directly is extremely difficult. Overcoming the accumulated error issue requires unreasonably robust models on both steps which, given the nature of low resource languages, is unlikely. However, it does not appear that augmentation of direct models with data from a closely related language suffers this issue to the same degree, showing modest increases in Bleu score across the board. In a similar vein, domain of the training set remains a major contributing factor to model inaccuracy, though this is unsurprising given the nature of neural networks.

While pushing the test data through a pivot translation step accumulates a lot of error, the same does not appear to be true of back translating a related language. Our data shows, in both the transformer and fine-tuned, pretrained sections, that creating new parallel data from the corpus of a related language/target language dataset by translating the related language into our source language largely mitigates the issue with accumulating error through multiple translation steps.

Also, we did some small scale experimentation with beam search. We didn't notice any significant increases in Bleu score while using it vs greed search, but we did experience a 20% decrease in training time.

Finally, on the subject of Bleu score, the use of Bleu score as a metric for judging the accuracy of translation is extremely limited. It may be (arguably) the best metric we have at the moment but that does not change that it is a poor representation of translation quality.

References

1. Singh, *Neural Machine Translation using Simple Seq2Seq Model*. 2020. https://medium.com/analytics-vidhya/neural-machine-translation-using-simple-seq2seq-517b5ce1ae0f
2. Sutskever, Vinyals, Le. *Sequence to Sequence Learning with Neural Networks*. 2014. https://arxiv.org/abs/1409.3215
3. Xia, Kong, Anastasopoulos, Neubig, *Generalized Data Augmentation for Low-Resource Translation.* 2019. https://www.aclweb.org/anthology/P19-1579.pdf
4. Wanshun Wong, *What is Teacher Forcing?*. 2019. https://towardsdatascience.com/what-is-teacher-forcing-3da6217fed1c
5. Jason Brownlee, *What is Teacher Forcing for Recurrent Neural Networks*. 2019. https://machinelearningmastery.com/teacher-forcing-for-recurrent-neural-networks/
6. PyTorch Documentation, *Sequence-to-Sequence Modeling with NN.Transformer and TorchText*. 2017. https://pytorch.org/tutorials/beginner/transformer_tutorial.html
7. Jay Alammar, *The Illustrated Transformer*. 2018. http://jalammar.github.io/illustrated-transformer/
8. Aladdin Persson, *Machine-Learning-Collection/.../seq2seq_transformer.py* 2020. https://github.com/aladdinpersson/Machine-Learning-Collection/blob/master/ML/Pytorch/more_advanced/seq2seq_transformer/seq2seq_transformer.py