Exploring Semi-Parametric NMT for Low-Resource Translation

Samuel Lurye Garrett Tanzer Alexander Wei

Abstract

In this paper, we apply state-of-the-art *semi-parametric* (or *retrieval-based*) NMT models to the recently formulated low-resource translation task of Guzmán et al. (2019). Retrieval-based approaches, which provide the sequence to sequence model with similar source/target pairs as context to help translate the source sentence in question, are outwardly well-suited to the low-resource translation task, because they excel in settings with multi-domain training data and test-time domain adaptation. However, we find that such techniques are ineffective on this task due to the particular domains involved. We explore precisely why this occurs and how the task differs from those where semi-parametric methods are effective.

1. Introduction

Modern neural machine translation (NMT) systems are rapidly approaching human-level performance for many language pairs (e.g., English–German). The success of these NMT models relies on the existence of vast corpora of parallel text with exact sentence-by-sentence translations. However, for language pairs with scant parallel data (e.g., English–Nepali), standard methods for NMT do not work nearly as well.

Existing approaches to low-resource translation include semi-supervised methods such as backtranslation (Sennrich et al., 2016), which incorporates data from monolingual corpora; weakly supervised methods (Xu & Koehn, 2017), which augment parallel datasets with noisier "comparable" sentences[1]; and fully unsupervised approaches (Lample & Conneau, 2019), which use no parallel data at all. However, none of these approaches perform well on the benchmark datasets constructed by Guzmán et al. (2019). These datasets highlight several challenging hurdles for low-resource translation: they are noisy, small in size, and feature a high degree of domain mismatch.

Our main contribution is to evaluate the effectiveness of *semi-parametric*, or *retrieval-based*, NMT techniques on the low-resource translation task, using the English–Nepali language pair from Guzmán et al. (2019). Semi-parametric techniques augment modern NMT models with *non-parametric* data, allowing the model to access parallel sentence pairs at test time that have similar context as the source. These kinds of models seem to be naturally motivated in the context of low-resource translation, which depends on small, noisy, and heterogeneous datasets: retrieval-based models are known to perform well on tasks with multi-domain training data, can adapt without parameter updates to different domains at test time, and are able to remember uncommon phrases (Bapna & Firat, 2019). In some sense, we can also think of the retrieval method as imposing a prior on the translation model, which can help to improve performance on small datasets.

However, our experiments show that these state-of-the-art retrieval-based methods are ineffective on the low-resource task, achieving a BLEU score of 0.67 on English to Nepali translation compared to the baseline of 4.19. We will explain why the model fails to adapt from the training set to the test set in this particular domain, even though it does so successfully in other problem settings, and share some of the other ways in which semi-parametric methods are surprisingly difficult to work with.

2. Background

One of the key challenges of low-resource translation is the scarcity of clean and relevant parallel data between genetically unrelated languages. As a point of comparison, the dataset provided by Guzmán et al. (2019) has only around 0.5M *noisy* sentence pairs, whereas the WMT' 14 datasets[2] have an order of magnitude more of *clean* data. Furthermore, low-resource translation data suffers from serious domain mismatch, where the training data (e.g., the Bible or Ubuntu user manuals) is very different from the data that needs to be translated. Finally, the source and target languages (e.g., English–Nepali) are highly dissimilar—both morphologically and syntactically. These issues compound to make translation a very difficult task for modern NMT systems. The best performing model of Guzmán et al. (2019) is a semi-supervised model that heavily uses backtranslation, yet it achieves less than 16 BLEU on translation from the

[1] http://statmt.org/paracrawl/

[2] https://www.statmt.org/wmt14/translation-task.html

low-resource language into English and less than 7 BLEU in the other direction.

Prior to Guzmán et al. (2019), there have not been many publically available datasets for low-resource translation. Thus researchers artifically restricted datasets for high-resource languages (e.g., English–French) to simulate the problem of low-resource translation. However, such datasets do not come with the same challenges of low-resource translation as noted above—the restricted data is still likely clean and relevant, and the language pairs themselves remain similar. Guzmán et al. (2019) address this by collating existing training data for English–Nepali and English–Sinhala translation and providing new, high-quality data for evaluation.

3. Related Work

A leading method for low-resource translation is *backtranslation* (Sennrich et al., 2016), a *semi-supervised* technique. The approach of backtranslation is to first train a (potentially low-quality) translation from the target language back to the source language. Then, one runs the model over a large monolingual corpus in the target language to get a large, but noisy parallel corpus. The key property of this parallel corpus is that the data of the target language is all native text. Thus, by training a NMT model on the clean parallel data augmented with the backtranslated data, the NMT is able to learn a language model for the target language on the backtranslated data, while learning how to translate on the given parallel data.

Another approach to low-resource translation is to augment the parallel data with noisier "comparable" data—the *weakly supervised* approach. The Paracrawl project provides large corpora of noisy comparable data obtained from crawling the web. It is also possible to augment the training data with a noisier dataset after applying a filtering algorithm (Xu & Koehn, 2017). Although not as high quality as actual parallel data, this noisier data can serve as a substitute when the data is particularly scarce, showing some small improvement in the experiments of (Xu & Koehn, 2017).

In addition to the related work in low-resource translation, relevant to our project is prior work in *semi-parametric translation*. There have been several papers exploring this approach in recent years, including (Bapna & Firat, 2019; Cao & Xiong, 2018; Gu et al., 2017).

Semi-parametric translation was introduced by Gu et al. (2017), and is a technique for machine translation that uses *retrieval* of similar sentences to augment the translation process. In addition to providing the model with a source sentence to be translated, semi-parametric translation provides the model with a set of relevant source–target sentence pairs that the model may use to aid its translation. In its original incarnation, this retrieval was done using a search engine, with sentences filtered by string similarity. Later, Cao & Xiong (2018) introduce retrieval based on sentence embeddings as well as a gating mechanism for using the retrieved contexts.

The work most directly related to our approach is the recent paper of Bapna & Firat (2019). They pioneer a semi-parametric approach for translation on English-French datasets that uses a pre-trained Transformer masked language model to encode sentences for non-parametric retrieval by n-gram encoding similarity, then uses a modified Transformer architecture to process retrieved nearest neighbors alongside the source sentence. We reimplement and apply their NMT architecture to the dataset of Guzmán et al. (2019). In particular, Bapna & Firat (2019)'s results appear to be strong motivation for semiparametric methods because they demonstrate the effectiveness of the techniques on multi-domain training sets and test-time domain adaptation, to the extent that it is more fruitful to use a particular training domain as a database for retrieval at test time than to include it in the training set itself.

3.1. Baseline

Our baseline comes from Guzmán et al. (2019), in which the authors present what they hope to become standardized evaluation datasets for the low-resource language pairs English–Nepali and English–Sinhala. We are primarily interested in the fully supervised setting for English–Nepali translation, whose baselines are shown in Table 1 below.

Src-Trg	BLEU
En–Ne	4.3
Ne–En	7.6

Table 1. Guzmán et al. (2019)'s baselines for fully supervised English–Nepali translation.

We base our comparison on the fully supervised setting because retrieval methods, although they may inject additional information through pretrained models like BERT, are orthogonal to existing weakly supervised or semi-supervised approaches, and can be applied in tandem with them. This distinction, however, is not especially important in light of how drastically semiparametric models underperform on this task.

4. Model

4.1. Semi-Parametric Translation

The fundamental problem we tackle is that of translation, which we formulate as follows: Let $\mathcal{V}_{\text{src}}$ be vocabulary of the source language and $\mathcal{V}_{\text{trg}}$ be the vocabulary of the target language. Our objective is to map using our model a sequence of tokens $X = (x_1, \ldots, x_n)$, $x_i \in \mathcal{V}_{\text{src}}$, to a sequence of tokens $Y = (y_1, \ldots, y_{n'})$, $y_i \in \mathcal{V}_{\text{trg}}$, such that the two sequences have the same "meaning."

As there no formal definition for "the same meaning", we will use the BLEU score (Papineni et al., 2001), which is largely a geometric average of n-gram precisions, to evaluate generated translations against a sample of handwritten translations. We will take as our objective maximizing the average BLEU score, as computed using SacreBLEU (Post, 2018), on sentences from our test set as our training objective.

However, as the BLEU score is a discrete function of the output, it is difficult to optimize over. Instead, optimization traditionally takes place on a conditional language modeling task in which the loss function is the *perplexity*:

$$\exp\left(-\frac{1}{n'}\sum_{i=1}^{n'}\log(h(y_i \mid y_{1:i-1}, x_{1:n}))\right).$$

In the above, $h(y_i \mid y_{1:i-1}, x_{1:n})$ is the predicted probability of y_i given the preceding tokens in the target sequence as well as the source sequence. Typically, h is modelled using a neural network architecture.

The above is the typical formulation of the translation problem for *parametric* translation. As we will be considering the *semi-parametric* setting, we will have the same optimization objective of perplexity, but our inputs will be augmented with a non-parametric context. That is, our model h will also have as an input a context $\Phi_X = ((X^1, Y^1), \ldots, (X^k, Y^k))$ of known sentence pairs from the source and target languages. Thus, the loss function for our training is

$$\exp\left(-\frac{1}{n'}\sum_{i=1}^{n'}\log(h(y_i \mid y_{1:i-1}, x_{1:n}, \Phi_X))\right).$$

The context Φ_X will be obtained from a separate retrieval model pretrained on monolingual data.

4.2. Model Details

For our project, we use the semi-parametric model proposed recently by Bapna & Firat (2019). This model is a neural machine translation model that takes as input both a source sentence and a non-parametrically retrieved list of sentence pairs. In this section, we will describe both the non-parametric retrieval process as well as the neural network architecture used to process the retrieved sentences.

4.2.1. Non-Parametric Retrieval

The first phase of the model is a non-parametric retrieval phase, in which for an input X, the model retrieves a context $\Phi_X = ((X^1, Y^1), \ldots, (X^k, Y^k))$, for some fixed context size k. Each element (X^i, Y^i) of the context is retrieved from a *retrieval set* $\mathcal{R}$ of parallel sentence pairs. To do the retrieval, we apply the *dense vector-based n-gram retrieval* process introduced by Bapna & Firat (2019).

We first use a BERT model (Devlin et al., 2018) pre-trained on the source language to generate contextual encodings for source sentence X of sentence pairs (X, Y) in $\mathcal{R}$. We then average these contextual encodings over windows of size 6, 12, and 18 to obtain contextual encodings for n-gram of these lengths in X. More specifically, for each length ℓ, we average over the contextual encodings of $(x_i, \ldots, x_{i+\ell-1})$, where $i \cong 1 \pmod{(n/2)}$. We store these n-gram contexts (and the sentence pair from which they came) in a database for retrieval.

To retrieve the non-parametric context from this database of n-gram encoding, we use nearest neighbor search in the ℓ_2 (i.e., Euclidean) metric. Given a source sentence X that we wish to translate, we apply the same processing as above to obtain n-gram contexts over n-grams of length 6, 12, and 18. For each n-gram, we find its nearest neighbors in the database along with the distances of the neighbors. We then sort the retrieved n-gram contexts by distance to the nearest n-gram context in X. Finally, we take the first k unique sentence pairs corresponding to elements in this sorted list. That is, we retrieve k sentence pairs by the metric that is the *minimum* of the pairwise distances between all n-gram contexts of X and the source sentence in the pair.

For intuition as to why this retrieval method makes sense, note that we are retrieving sentence pairs that are likely to share similar phrases between the source sentence and X. This is the purpose of n-gram-level similarity as opposed to sentence-level similarity. We expect this to provide useful phrase-level context through the target sentence of the pair that could be used to aid translation.

We implement our retrieval stage with FAISS, an efficient (approximate) nearest neighbor library implemented by the authors of (Johnson et al., 2017).

4.2.2. Neural Machine Translation with a Retrieved Context

The second phase of the model uses the retrieved context Φ_X as part of the translation network. The architecture we used here was introduced recently by Bapna & Firat (2019).

The backbone of our model is the Transformer architecture (Vaswani et al., 2017). Our sequence to sequence model consists of an encoder and then a decoder, each comprised of stacked Transformer attention layers. The Transformer attention layer in each encoder is a self-attention layer, where keys, values, and queries for attention are each the contexts of elements in the sequence. On the other hand, the decoder attention layers consist of self-attention layers interleaved with cross-attention layers, where in the latter, keys and values are encoder contexts and queries are decoder contexts.

Conditional Source Target Memory. To adapt this model to the semi-parametric setting, we use a variant of Transformer-style encoding on the retrieved output to create a *conditional source target memory* (CSTM). The CSTM is designed to be a vector, contructed using attention, that summarizes all of the relevant context from the retrieved sentence pairs for decoding.

Let X be our source sentence and (X^i, Y^i) be the retrieved sentence pairs. For each retrieved sentence pair (X^i, Y^i), we do the following:

- First, encode X with a single Transformer self-attention layer.
- Next, encode X^i with a self-attention layer followed by cross-attention with the encoding of X as keys and values and the encoding of X^i as queries. (This is similar to a Transformer decoder layer.)
- Finally, encode Y^i with a self-attention layer followed by a cross-attention layer with the output of the previous step as keys and values and the encoding of Y^i as the queries.

The output of these encoding steps for each retrieved are concatenated to form CSTM. For a diagrammatic depiction, see Figure 1.

Gated Multi-Source Attention Given the CSTM, it still remains to combine the CSTM (which consists of contexts obtained from the retrieved sentence pairs *conditioned* on the source sentence) with the source sentence itself to create a translation through a decoder. For this, we use *gated multi-source attention*, which was first introduced in (Cao & Xiong, 2018). The intention behind gated multi-source attention is to allow the model to learn *when* to use the CSTM versus the context of the encoded source.

In gated multi-source attention, we first encode the source with a Transformer encoder. Then, we apply a Transformer decoder network, such that in each cross-attention layer, we attend over both the encoded source and the CSTM. Let c_s and c_m be the context vectors obtained from the source and CSTM, respectively. Given these context vectors, we combine them into a single gated context vector c with a gating variable

$$g = \sigma(W_s c_s + W_m c_m)$$

obtained from a feedforward layer; i.e., the combined context is

$$c = g \circ c_s + (1 - g) \circ c_m,$$

where $\circ$ denotes element-wise multiplication. Note that W_s and W_m in this case are learned parameters of the model. For a diagrammatic depiction, see Figure 2.

4.3. Model Summary

To summarize, the model consists of a pre-trained retrieval component combined with the NMT architecture of Bapna & Firat (2019). The NMT architecture follows the Transformer encoder-decoder architecture, but with a modified encoder that outputs a CSTM from the retrieved sentence pairs. This CSTM is intended to provided a bank of information about similar phrases and sentences for the encoder to look up. The decoder is implemented with gated multi-source attention over the encoded source and the CSTM. Thus, the model consists of the parameters in each of the Transformer attention layers along with recombination weights at the gating layer—we optimize over all of these parameters using adaptive stochastic gradient descent.

5. Training

Optimizing our model is relatively straightforward because it is fully differentiable. We minimize perplexity on the semi-parametric conditional language modeling task, as described in Section 4.1, using backpropagation and the Adam optimizer to determine parameter updates. Note that, as described in Section 4.2.1, we precompute BERT encodings for a reduced set of n-grams in the retrieval set and store them in a database, then retrieve the (approximate) nearest neighbors from this set as a preprocessing step that occurs once before training. This portion of our model has no parameters and is not optimized; only the Transformer encoder-decoder architecture and CSTM module receive parameter updates during training.

6. Methods

Our dataset, as provided by Guzmán et al. (2019), consists of 563,254 parallel training sentences, 2,559 parallel validation sentences, and 2,835 parallel test sentences. Of the training set, 495K sentences come from the GNOME/KDE/Ubuntu handbooks, 62K come from the Bible, 4K come from the Penn Tree Bank dataset, and 3K come from the Global Voices dataset. Although the number of sentences is not particularly small, the quality overall is very poor. For example, the GNOME/KDE/Ubuntu sentences are very short, totalling only 2M tokens, and are exceedingly domain specific, which means they are of questionable use in building an effective translation system.

As in Guzmán et al. (2019), we learn 5000 byte-pair encoding tokens that are shared between the source and target languages with SentencePiece, and use these to tokenize the data. We perform retrieval entirely offline, as described

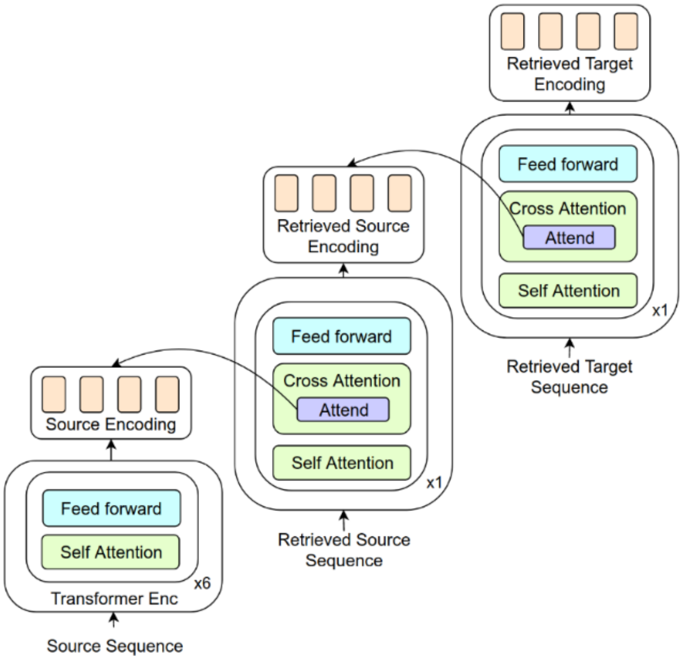

Figure 1. Depiction of CSTM construction from Bapna & Firat (2019).

in Section 4.2.1, and augment the dataset with retrieved source/target pairs before training on it.

Because Bapna & Firat (2019)'s model is not publicly available, we reimplemented the architecture with the Fairseq Python library (Ott et al., 2019), which is used for (Bapna & Firat, 2019)'s baseline. Although Fairseq comes with a built-in Transformer architecture, extending the functionality to include what we needed for our model required a relatively substantial amount of code.

The architectures of the main encoder and decoder Transformers are identical to those used in Guzmán et al. (2019). The two Transformers used to encode the retrieved source and retrieved target sentences each have only one transformer layer, as in Bapna & Firat (2019), but otherwise have the same dimensions and settings as the main encoder and decoder. Additionally, the one-layer Transformers share their embeddings with the main Transformers.

Our only optimization settings and hyperparameters that differ from Guzmán et al. (2019) are the batch size of at most 10K tokens (due to memory constraints) and the learning rate of 10^{-4}, which is necessary for handling the smaller batch size with half-precision floats. Despite the different learning rate, we use the same learning rate schedule they do, namely that of Ott et al. (2018).

We focused on only the English to Nepali direction in order to concentrate our computational resources. In order to provide a sound comparison, we replicated Guzmán et al. (2019)'s baseline for translation in this direction locally, using their publicly available code. We used their hyperparameter settings and trained for 150 epochs. We only trained Bapna & Firat (2019)'s model for 20 epochs, due to the substantially increased computational cost and the observation that it was considerably underperforming the baseline model in a like-for-like comparison. Moreover, limited memory meant that we could retrieve at most 4 nearest neighbors per sentence. We used a single Tesla T4 GPU to train each model; a single epoch took about 5 minutes for the baseline and 30 minutes for the retrieval model.

To produce translations for evaluation with SacreBLEU, we used beam search with beam width 5 for the baseline and beam width 2 for the Bapna & Firat (2019) model. For the latter, a larger beam width used too much memory.

7. Results

Our replicated baseline for English to Nepali achieved a test BLEU of 4.19, compared to the stated baseline of 4.3. Because the retrieval-based model uses the validation set at test time, to be generous we also trained a baseline model that includes the validation set in its training set, which surprisingly achieved a lower score of 3.93. These results are shown in Table 2.

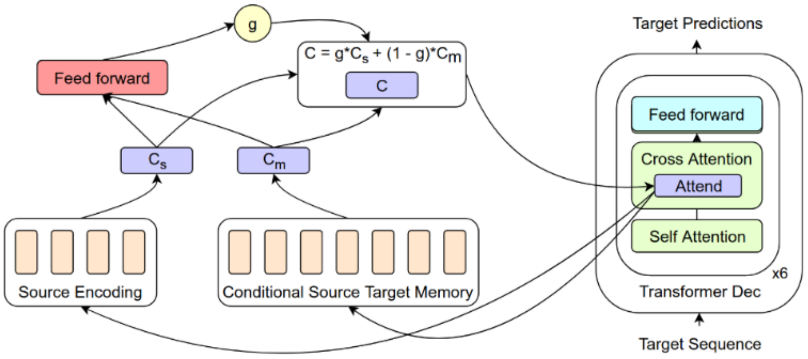

Figure 2. Depiction of gated multi-source attention from Bapna & Firat (2019).

Baseline training set	Val BLEU	Test BLEU
Train	2.79	4.19
Train + Val	2.66	3.93

Table 2. BLEU scores on the validation set and test set, for *baselines* trained on the training set alone and the combined training + validation set.

We trained several configurations of our semi-parametric model using different datasets as the retrieval set; these results are shown in Table 3. The best-performing variant was the one that, for each sentence in the training set, retrieved only sentences from the validation, achieving a test BLEU of 0.67. The variants that retrieved from only the training set and from both the training set and validation set were effectively tied, at 0.19 and 0.18 BLEU respectively. Interestingly, the model that retrieved from both the train and the validation sets scored a validation BLEU of 8.20, indicating that information from the retrieved validation sentences was being stored in the model, despite this not happening when sentences from *only* the validation set were retrieved.

Retrieval training database	Val BLEU	Test BLEU
Train	0.16	0.19
Train + Val	8.20	0.18
Val	0.62	0.67

Table 3. BLEU scores on the validation set and test set, for *retrieval-based models* where training examples had nearest neighbors retrieved from the training set only; both the training set and the validation set; and the validation set only.

8. Discussion

Though—or perhaps because—our results are negative, we have much to discuss. The first surprising result is that the baseline we trained on the train and validation sets together performs worse than the bsaeline trained on the training set alone, as seen in Table 2. While this might be due to random initialization, one would expect the former model to perform substantially better on the validation set, even if not the test set, but this was not the case. This relates to Bapna & Firat (2019)'s discussion of multi-domain training, where the presence of a domain in the training set is not necessarily enough to ensure good performance on it, especially when the proportion of examples in the training set is skewed towards one domain or the other.

Yet again, this phenomenon suggests that retrieval-based models, which are known to be effective for multi-domain training and test-time domain adaptation, would be a natural fit here. But we found that the best model of this kind performs miserably on this task, as seen in Table 3. In our quest to explore the performance tradeoff curve by decreasing the number of retrieved sentences k, we actually found that doing so improved the performance of the model, especially in the limit $k = 0$, where performance shot back up to that of the baseline (because it reduces to the same Transformer architecture). Likewise, the model that performs the best is the one that retrieves the least relevant sentences during training: the one that retrieves only from the validation set achieves a test BLEU of 0.67. We can see in Figure 3 that the attention visualizations from this model do not use the source/target context in a focused way; most of the attention is diffuse, and high gate values do not pattern with rows that have sharp activations. In virtually every example, the sharpest alignment comes from the `<end>` token in each sentence.

So why is this the case? We believe that this discrepancy comes from the fact that the training set domains are much more homogeneous than the validation and test sets. We can see this empirically in Table 4, which shows that the average sentence in the training set is 3.762 units away from its nearest neighbor in BERT encoding space, compared to 38.567 for the test set—an order of magnitude difference. This means that when we allow the model to train using retrieval on the train set, as in the model that scores 0.19 in Table 3, it learns to copy from the retrieved sentences much more readily than if we had a training set with less homogeneous sentences. In particular, a large portion of the training set, the GNOME data, also consists of short and often repeated or redundant "sentences", meaning that the retrieved sentences may fully specify the translation.

Dataset	Average ℓ_2 distance
Train	3.762
Val	38.920
Test	38.567

Table 4. Average ℓ_2 distance between the BERT encoding of a sentence and its nearest neighbor, across dataset splits.

We observe this happening in practice through the training perplexity and loss of any model that retrieves sentences from the training set; within about an epoch, it drops to 10ppl, whereas the validation-only-retrieval model or the baseline would be at about 1000ppl. Because copying can fully account for this data's translation, we are effectively losing all of the information in this domain of the training set (while still incurring an enormous computational cost each time through the training loop), because the model has no incentive to learn this information in its own weights, making this an ineffective way to filter out noisy data. This problem is only compounded by the fact that when we run the model on longer sentences, we do not retrieve more neighbors, meaning that proportionally more information must come from the model itself. Another surprising result that fits into this regime is the validation BLEU score for the model that retrieves from both the train and validation set—8.20—which far exceeds the 0.62 score of the model using only the validation sentences. We believe that information flows more readily into the model weights from the retrieved validation sentences in the former case because there is less pressure for those weights to be used for other purposes—all the model needs to do to minimize the training loss is to copy from the retrieved sentences. Meanwhile, in the latter model with only validation sentences, the model weights need to actually learn to translate the training data without the pattern of retrieved train sentences.

Finally, we discuss the computational implications of using a retrieval-based model. Because the retrieval itself is performed offline as dataset augmentation, this is actually not the significant cost. Rather, the extra memory and compute units required to run k additional sentences through an encoders and attention for each example and store the intermediates for backpropagation is a crippling weakness, which affects model size, batch size, and even beam width for beam search; we were only able to use a beam width of 2 for the retrieval model, compared to 5 for the baseline, and on the baseline this difference amounted to a BLEU difference of about 0.3. This apparent k times overhead is actually exacerbated by the fact that we cannot batch retrieved sentences by length, like we normally can by scheduling sentences during training. We see in Figure 4 that even in the most generous comparison, where we look at the baseline and retrieval model across epochs, despite the fact that the retrieval model is churning through k times as much data and takes more wall clock time, the baseline is strictly better.

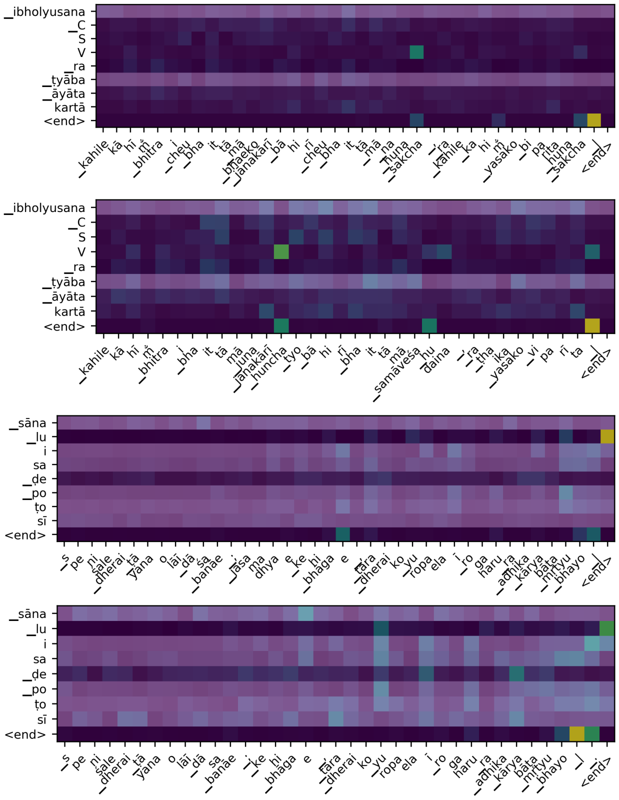

Figure 3. Sample attention activations from the bottom layer of the CSTM, with the retrieved sentence horizontally and the goal sentence vertically. Bright colors indicate focused attention, and light horizontal lines indicate a high gate value.

9. Conclusion

We applied techniques in semi-parametric NMT, primarily those due to Bapna & Firat (2019), to the low-resource translation task formulated in Guzmán et al. (2019). Our best retrieval-based model achieved a test BLEU score of 0.67 on translation from English to Nepali, compared to our reproduced baseline, which reached 4.19. But in fact, this

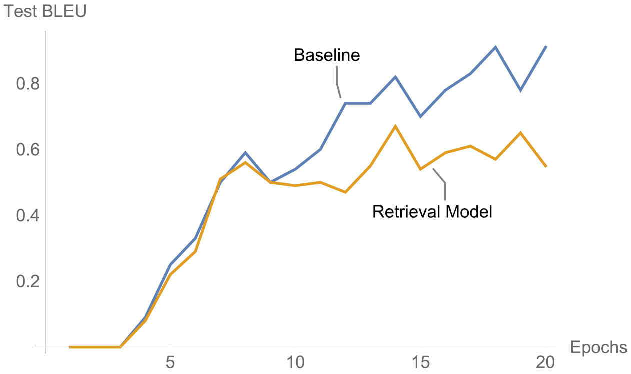

Figure 4. Test BLEU across number of epochs for our model and the baseline with equal hyperparameter settings.

best model was the one that utilized retrieval the *least*: it used few, irrelevant retrieved sentences and apparently did not pick out any salient features in the retrieved source/target attention activations.

We believe that this is because this task is fundamentally unsuited to retrieval-based methods, because the different domains have different self-similarity characteristics. A large portion of the training data is composed of "sentences" of one or two words, meaning that one or two retrieved sentences are close enough to exactly contain a suitable translation and obviating the need for the model to learn anything but copying. We confirmed this observation empirically with the average ℓ_2 distance to a nearest neighbor across domains. We think that in order for domain adaptation in semi-parametric methods to work, the domains must have similar distributions with respect to retrieval.

On top of these concerns for ultimate model accuracy, semi-parametric models have substantial practical costs, namely the increased memory usage and computational cost of encoding and attending to k extra sentences for each input. This cost is only exacerbated by the fact that retrieved sentences cannot easily be batched by length in the same way as traditional input sentences, inflating the amount of padding and reducing GPU utilization.

Given the success of Bapna & Firat (2019), semi-parametric methods have their place in the natural language processing pantheon, but that place is not low-resource translation.

References

Bapna, Ankur and Firat, Orhan. Non-parametric adaptation for neural machine translation. *CoRR*, abs/1903.00058, 2019.

Cao, Qian and Xiong, Deyi. Encoding gated translation memory into neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pp. 3042–3047, 2018.

Devlin, Jacob, Chang, Ming-Wei, Lee, Kenton, and Toutanova, Kristina. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Gu, Jiatao, Wang, Yong, Cho, Kyunghyun, and Li, Victor O. K. Search engine guided non-parametric neural machine translation. *CoRR*, abs/1705.07267, 2017.

Guzmán, Francisco, Chen, Peng-Jen, Ott, Myle, Pino, Juan, Lample, Guillaume, Koehn, Philipp, Chaudhary, Vishrav, and Ranzato, Marc'Aurelio. Two new evaluation datasets for low-resource machine translation: Nepali-english and sinhala-english. *CoRR*, abs/1902.01382, 2019.

Johnson, Jeff, Douze, Matthijs, and Jégou, Hervé. Billion-scale similarity search with gpus. *arXiv preprint arXiv:1702.08734*, 2017.

Lample, Guillaume and Conneau, Alexis. Cross-lingual language model pretraining. *CoRR*, abs/1901.07291, 2019. URL http://arxiv.org/abs/1901.07291.

Ott, Myle, Edunov, Sergey, Grangier, David, and Auli, Michael. Scaling neural machine translation. In *Proceedings of the Third Conference on Machine Translation: Research Papers, WMT 2018, Belgium, Brussels, October 31 - November 1, 2018*, pp. 1–9, 2018.

Ott, Myle, Edunov, Sergey, Baevski, Alexei, Fan, Angela, Gross, Sam, Ng, Nathan, Grangier, David, and Auli, Michael. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*, 2019.

Papineni, Kishore, Roukos, Salim, Ward, Todd, and Zhu, Wei-Jing. Bleu: a method for automatic evaluation of machine translation. In *ACL*, 2001.

Post, Matt. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers, WMT 2018, Belgium, Brussels, October 31 - November 1, 2018*, pp. 186–191, 2018.

Sennrich, Rico, Haddow, Barry, and Birch, Alexandra. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*, 2016.

Vaswani, Ashish, Shazeer, Noam, Parmar, Niki, Uszkoreit, Jakob, Jones, Llion, Gomez, Aidan N., Kaiser, Lukasz, and Polosukhin, Illia. Attention is all you need. In *Advances in Neural Information Processing Systems 30:*

Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA, pp. 6000–6010, 2017.

Xu, Hainan and Koehn, Philipp. Zipporah: a fast and scalable data cleaning system for noisy web-crawled parallel corpora. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017*, pp. 2945–2950, 2017.

10. Appendix

10.1. Code

Our fork of Fairseq is available here. We made the following changes:

- Created `fairseq/models/cstm_transformer.py`: this contains the bulk of the code for our model
- Modified `fairseq/data/language_pair_dataset.py`: added class `LanguagePairDatasetWithIndex`
- Modified `fairseq/tasks/translation.py`: added class `CSTMTranslationTask`
- Modified `generate.py`: added a few lines to load all dataset splits when generating translations with the CSTM model

Code for running our model is available on this Colab notebook. The preprocessed data is available on Google Drive here and can be fed directly into the model in this form.