Correcting Length Bias in Neural Machine Translation

Abstract

We study two problems in neural machine translation (NMT). First, in beam search, whereas a wider beam should in principle help translation, it often hurts NMT. Second, NMT has a tendency to produce translations that are too short. Here, we argue that these problems are closely related and both rooted in label bias. We show that correcting the brevity problem almost eliminates the beam problem; we compare some commonly-used methods for doing this, finding that a simple per-word reward works well; and we introduce a simple and quick way to tune this reward using the perceptron algorithm.

Introduction

Although highly successful, neural machine translation (NMT) systems continue to be plagued by a number of problems. We focus on two here: the beam problem and the brevity problem.

First, machine translation systems rely on heuristics to search through the intractably large space of possible translations. Most commonly, beam search is used during the decoding process. Traditional statistical machine translation systems often rely on large beams to find good translations. However, in neural machine translation, increasing the beam size has been shown to degrade performance. This is the last of the six challenges identified by BIBREF0 .

The second problem, noted by several authors, is that NMT tends to generate translations that are too short. BIBREF1 and BIBREF0 address this by dividing translation scores by their length, inspired by work on audio chords BIBREF2 . A similar method is also used by Google's production system BIBREF3 . A third simple method used by various authors BIBREF4 , BIBREF5 , BIBREF6 is a tunable reward added

for each output word. BIBREF7 and BIBREF8 propose variations of this reward that enable better guarantees during search.

In this paper, we argue that these two problems are related (as hinted at by BIBREF0) and that both stem from label bias, an undesirable property of models that generate sentences word by word instead of all at once.

The typical solution is to introduce a sentence-level correction to the model. We show that making such a correction almost completely eliminates the beam problem. We compare two commonly-used corrections, length normalization and a word reward, and show that the word reward is slightly better.

Finally, instead of tuning the word reward using grid search, we introduce a way to learn it using a perceptron-like tuning method. We show that the optimal value is sensitive both to task and beam size, implying that it is important to tune for every model trained. Fortunately, tuning is a quick post-training step.

Problem

Current neural machine translation models are examples of locally normalized models, which estimate the probability of generating an output sequence INLINEFORM0 as INLINEFORM1

For any partial output sequence INLINEFORM0 , let us call INLINEFORM1 , where INLINEFORM2 ranges over all possible completions of INLINEFORM3 , the suffix distribution of INLINEFORM4 . The suffix distribution must sum to one, so if the model overestimates INLINEFORM5 , there is no way for the suffix distribution to downgrade it. This is known as label bias BIBREF9 , BIBREF10 .

Label bias in sequence labeling

Label bias was originally identified in the context of HMMs and MEMMs for sequence-labeling tasks, where the input sequence INLINEFORM0 and output sequence INLINEFORM1 have the same length, and INLINEFORM2 is conditioned only on the partial input sequence INLINEFORM3 . In this case, since INLINEFORM4 has no knowledge of future inputs, it's much more likely to be incorrectly estimated. For example, suppose we had to translate, word-by-word, un hélicoptère to a helicopter (Figure FIGREF2). Given just the partial input un, there is no way to know whether to translate it as a or an. Therefore, the probability for the incorrect translation INLINEFORM5 will turn out to be an overestimate. As a result, the model will overweight translations beginning with an, regardless of the next input word.

This effect is most noticeable when the suffix distribution has low entropy, because even when new input (hélicoptère) is revealed, the model will tend to ignore it. For example, suppose that the available translations for hélicoptère are helicopter, chopper, whirlybird, and autogyro. The partial translation a must divide its probability mass among the three translations that start with a consonant, while an gives all its probability mass to autogyro, causing the incorrect translation an autogyro to end up with the highest probability.

In this example, INLINEFORM0 , even though overestimated, is still lower than INLINEFORM1 , and wins only because its suffixes have higher probability. Greedy search would prune the incorrect prefix an and yield the correct output. In general, then, we might expect greedy or beam search to alleviate some symptoms of label bias. Namely, a prefix with a low-entropy suffix distribution can be pruned if its probability is, even though overestimated, not among the highest probabilities. Such an observation was made by BIBREF11 in the context of dependency parsing, and we will see next that precisely such a situation affects output length in NMT.

Length bias in NMT

In NMT, unlike the word-by-word translation example in the previous section, each output symbol is conditioned on the entire input sequence. Nevertheless, it's still possible to overestimate or underestimate INLINEFORM0 , so the possibility of label bias still exists. We expect that it will be more visible with weaker models, that is, with less training data.

Moreover, in NMT, the output sequence is of variable length, and generation of the output sequence stops when </s> is generated. In effect, for any prefix ending with </s>, the suffix distribution has zero entropy. This situation parallels example of the previous section closely: if the model overestimates the probability of outputting </s>, it may proceed to ignore the rest of the input and generate a truncated translation.

Figure FIGREF4 illustrates how this can happen. Although the model can learn not to prefer shorter translations by predicting a low probability for INLINEFORM0 early on, at each time step, the score of INLINEFORM1 puts a limit on the total remaining score a translation can have; in the figure, the empty translation has score INLINEFORM2 , so that no translation can have score lower than INLINEFORM3 . This lays a heavy burden on the model to correctly guess the total score of the whole translation at the outset.

As in our label-bias example, greedy search would prune the incorrect empty translation. More generally, consider beam search: at time step INLINEFORM0 , only the top INLINEFORM1 partial or complete translations are retained while the rest are pruned. (Implementations of beam search vary in the details, but this variant is simplest for the sake of argument.) Even if a translation ending at time INLINEFORM2 scores higher than a longer translation, as long as it does not fall within the top INLINEFORM3 when compared with partial translations of length INLINEFORM4 (or complete translations of length at most

INLINEFORM5), it will be pruned and unable to block the longer translation. But if we widen the beam (INLINEFORM6), then translation accuracy will suffer. We call this problem (which is BIBREF0 's sixth challenge) the beam problem. Our claim, hinted at by BIBREF0 , is that the brevity problem and the beam problem are essentially the same, and that solving one will solve the other.

Correcting Length

To address the brevity problem, many designers of NMT systems add corrections to the model. These corrections are often presented as modifications to the search procedure. But, in our view, the brevity problem is essentially a modeling problem, and these corrections should be seen as modifications to the model (Section SECREF5). Furthermore, since the root of the problem is local normalization, our view is that these modifications should be trained as globally-normalized models (Section SECREF6).

Models

Without any length correction, the standard model score (higher is better) is: INLINEFORM0

To our knowledge, there are three methods in common use for adjusting the model to favor longer sentences.

Length normalization divides the score by INLINEFORM0 BIBREF0 , BIBREF1 , BIBREF2 : INLINEFORM1

Google's NMT system BIBREF3 relies on a more complicated correction: INLINEFORM0

Finally, some systems add a constant word reward BIBREF5 : INLINEFORM0

If INLINEFORM0 , this reduces to the baseline model. The advantage of this simple reward is that it can be computed on partial translations, making it easier to integrate into beam search.

Training

All of the above modifications can be viewed as modifications to the base model so that it is no longer a locally-normalized probability model.

To train this model, in principle, we should use something like the globally-normalized negative log-likelihood: INLINEFORM0

where INLINEFORM0 is the reference translation. However, optimizing this is expensive, as it requires performing inference on every training example or heuristic approximations BIBREF12 , BIBREF13 .

Alternatively, we can adopt a two-tiered model, familiar from phrase-based translation BIBREF4 , first training INLINEFORM0 and then training INLINEFORM1 while keeping the parameters of INLINEFORM2 fixed, possibly on a smaller dataset. A variety of methods, like minimum error rate training BIBREF14 , BIBREF5 , are possible, but keeping with the globally-normalized negative log-likelihood, we obtain, for the constant word reward, the gradient: INLINEFORM3

where INLINEFORM0 is the 1-best translation. Then the stochastic gradient descent update is just the familiar perceptron rule: INLINEFORM1

although below, we update on a batch of sentences rather than a single sentence. Since there is only one parameter to train, we can train it on a relatively small dataset.

Length normalization does not have any additional parameters, with the result (in our opinion, strange) that a change is made to the model without any corresponding change to training. We could use gradient-based methods to tune the INLINEFORM0 in the GNMT correction, but the perceptron approximation turns out to drive INLINEFORM1 to INLINEFORM2 , so a different method would be needed.

Experiments

We compare the above methods in four settings, a high-resource German–English system, a medium-resource Russian–English system, and two low-resource French–English and English–French systems. For all settings, we show that larger beams lead to large BLEU and METEOR drops if not corrected. We also show that the optimal parameters can depend on the task, language pair, training data size, as well as the beam size. These values can affect performance strongly.

Data and settings

Most of the experimental settings below follow the recommendations of BIBREF15 . Our high-resource, German–English data is from the 2016 WMT shared task BIBREF16 . We use a bidirectional encoder-decoder model with attention BIBREF17 . Our word representation layer has 512 hidden units, while other hidden layers have 1024 nodes. Our model is trained using Adam with a learning rate of 0.0002. We use 32k byte-pair encoding (BPE) operations learned on the combined source and target training data BIBREF19 . We train on minibatches of size 2012 words and validate every 100k sentences, selecting the final model based on development perplexity. Our medium-resource, Russian–English system uses data from the 2017 WMT translation task, which consists of roughly 1 million training sentences BIBREF20 . We use the same architecture as our German–English system, but only have 512 nodes in all layers. We use 16k BPE operations and dropout of 0.2. We train on minibatches of 512 words

and validate every 50k sentences.

Our low-resource systems use French and English data from the 2010 IWSLT TALK shared task BIBREF21 . We build both French–English and English–French systems. These networks are the same as for the medium Russian-English task, but use only 6k BPE operations. We train on minibatches of 512 words and validate every 30k sentences, restarting Adam when the development perplexity goes up.

To tune our correction parameters, we use 1000 sentences from the German–English development dataset, 1000 sentences from the Russian–English development dataset, and the entire development dataset for French–English (892 sentences). We initialize the parameter, INLINEFORM0 . We use batch gradient descent, which we found to be much more stable than stochastic gradient descent, and use a learning rate of INLINEFORM1 , clipping gradients for INLINEFORM2 to 0.5. Training stops if all parameters have an update of less than 0.03 or a max of 25 epochs was reached.

Solving the length problem solves the beam problem

Here, we first show that the beam problem is indeed the brevity problem. We then demonstrate that solving the length problem does solve the beam problem. Tables TABREF10 , TABREF11 , and TABREF12 show the results of our German–English, Russian–English, and French–English systems respectively. Each table looks at the impact on BLEU, METEOR, and the ratio of the lengths of generated sentences compared to the gold lengths BIBREF22 , BIBREF23 . The baseline method is a standard model without any length correction. The reward method is the tuned constant word reward discussed in the previous section. Norm refers to the normalization method, where a hypothesis' score is divided by its length.

The top sections of Tables TABREF10 , TABREF11 , TABREF12 illustrate the brevity and beam

problems in the baseline models. As beam size increases, the BLEU and METEOR scores drop significantly. This is due to the brevity problem, which is illustrated by the length ratio numbers that also drop with increased beam size. For larger beam sizes, the length of the generated output sentences are a fraction of the lengths of the correct translations. For the lower-resource French–English task, the drop is more than 8 BLEU when increasing the beam size from 10 to 150. The issue is even more evident in our Russian-English system where we increase the beam to 1000 and BLEU scores drop by more than 20 points.

The results of tuning the word reward, INLINEFORM0 , as described in Section SECREF6 , is shown in the second section of Tables TABREF10 , TABREF11 , and TABREF12 . In contrast to our baseline systems, our tuned word reward always fixes the brevity problem (length ratios are approximately 1.0), and generally fixes the beam problem. An optimized word reward score always leads to improvements in METEOR scores over any of the best baselines. Across all language pairs, reward and norm have close METEOR scores, though the reward method wins out slightly. BLEU scores for reward and norm also increase over the baseline in most cases, despite BLEU's inherent bias towards shorter sentences. Most notably, whereas the baseline Russian–English system lost more than 20 BLEU points when the beam was increased to 1000, our tuned reward score resulted in a BLEU gain over any baseline beam size. Whereas in our baseline systems, the length ratio decreases with larger beam sizes, our tuned word reward results in length ratios of nearly 1.0 across all language pairs, mitigating many of the issues of the brevity problem.

We note that the beam problem in NMT exists for relatively small beam sizes – especially when compared to traditional beam sizes in SMT systems. On our medium-resource Russian–English system, we investigate the full impact of this problem using a much larger beam size of 1000. In Table TABREF10 , we can see that the beam problem is particularly pronounced. The first row of the table shows the uncorrected, baseline score. From a beam of 10 to a beam of 1000, the drop in BLEU scores is over 20

points. This is largely due to the brevity problem discussed earlier. The second row of the table shows the length of the translated outputs compared to the lengths of the correct translations. Though the problem persists even at a beam size of 10, at a beam size of 1000, our baseline system generates less than one third the number of words that are in the correct translations. Furthermore, 37.3% of our translated outputs have sentences of length 0. In other words, the most likely translation is to immediately generate the stop symbol. This is the problem visualized in Figure FIGREF4 .

However, when we tune our word reward score with a beam of 1000, the problem mostly goes away. Over the uncorrected baseline, we see a 22.0 BLEU point difference for a beam of 1000. Over the uncorrected baseline with a beam of 10, the corrected beam of 1000 gets a BLEU gain of 0.8 BLEU. However, the beam of 1000 still sees a drop of less than 1.0 BLEU over the best corrected version. The word reward method beats the uncorrected baseline and the length normalization correction in almost all cases.

Another way to demonstrate that the beam problem is the same as the brevity problem is to look at the translations generated by baseline systems on shorter sentences. Figure FIGREF18 shows the BLEU scores of the Russian–English system for beams of size 10 and 1000 on sentences of varying lengths, with and without correcting lengths. The x-axes of the figure are cumulative: length 20 includes sentences of length 0–20, while length 10 includes 0–10. It is worth noting that BLEU is a word-level metric, but the systems were built using BPE; so the sequences actually generated are longer than the x-axes would suggest.

The baseline system on sentences with 10 words or less still has relatively high BLEU scores—even for a beam of 1000. Though there is a slight drop in BLEU (less than 2), it is not nearly as severe as when looking at the entire test set (more than 20). When correcting for length with normalization or word reward, the problem nearly disappears when considering the entire test set, with reward doing slightly

better. For comparison, the rightmost points in each of the subplots correspond to the BLEU scores in columns 10 and 1000 of Table TABREF10 . This suggests that the beam problem is strongly related to the brevity problem.

The interaction between the length problem and the beam problem can be visualized in the histograms of Figure FIGREF19 on the Russian–English system. In the upper left plot, the uncorrected model with beam 10 has the majority of the generated sentences with a length ratio close to 1.0, the gold lengths. Going down the column, as the beam size increases, the distribution of length ratios skews closer to 0. By a beam size of 1000, 37% of the sentences have a length of 0. However, both the word reward and the normalized models remain very peaked around a length ratio of 1.0 even as the beam size increases.

Tuning word reward

Above, we have shown that fixing the length problem with a word reward score fixes the beam problem. However these results are contingent upon choosing an adequate word reward score, which we have done in our experiments by optimization using a perceptron loss. Here, we show the sensitivity of systems to the value of this penalty, as well as the fact that there is not one correct penalty for all tasks. It is dependent on a myriad of factors including, beam size, dataset, and language pair.

In order to investigate how sensitive a system is to the reward score, we varied values of INLINEFORM0 from 0 to INLINEFORM1 on both our German–English and Russian–English systems with a beam size of 50. BLEU scores and length ratios on 1000 heldout development sentences are shown in Figure FIGREF27 . The length ratio is correlated with the word reward as expected, and the BLEU score varies by more than 5 points for German–English and over 4.5 points for Russian–English. On German–English, our method found a value of INLINEFORM2 , which is slightly higher than optimal; this is because the heldout sentences have a slightly shorter length ratio than the training sentences. Conversely, on

Russian–English, our found value of INLINEFORM3 is slightly lower than optimal as these heldout sentences have a slightly higher length ratio than the sentences used in training.

Tuning the reward penalty using the method described in Section SECREF6 resulted in consistent improvements in METEOR scores and length ratios across all of our systems and language pairs. Tables TABREF10 , TABREF11 , and TABREF12 show the optimized value of INLINEFORM0 for each beam size. Within a language pair, the optimal value of INLINEFORM1 is different for every beam size. Likewise, for a given beam size, the optimal value is different for every system. Our French–English and English–French systems in Table TABREF12 have the exact same architecture, data, and training criteria. Yet, even for the same beam size, the tuned word reward scores are very different.

Low-resource neural machine translation performs significantly worse than high-resource machine translation BIBREF0 . Table TABREF26 looks at the impact of training data size on BLEU scores and the beam problem by using 10% and 50% of the available Russian–English data. Once again, the optimal value of INLINEFORM0 is different across all systems and beam sizes. Interestingly, as the amount of training data decreases, the gains in BLEU using a tuned reward penalty increase with larger beam sizes. This suggests that the beam problem is more prevalent in lower-resource settings, likely due to the fact that less training data can increase the effects of label bias.

Fortunately, the tuning process is very inexpensive. Although it requires decoding on a development dataset multiple times, we only need a small dataset. The time required for tuning our French–English and German–English systems is shown in Table TABREF13 . These experiments were run on an Nvidia GeForce GTX 1080Ti. The tuning usually takes a few minutes to hours, which is just a fraction of the overall training time. We note that there are numerous optimizations that could be taken to speed this up even more, such as storing the decoding lattice for partial reuse. However, we leave this for future work.

Word reward vs. length normalization

Tuning the word reward score generally had higher METEOR scores than length normalization across all of our settings. With BLEU, length normalization beat the word reward on German-English and French–English, but tied on English-French and lost on Russian–English. For the largest beam of 1000, the tuned word reward had a higher BLEU than length normalization. Overall, the two methods have relatively similar performance, but the tuned word reward has the more theoretically justified, globally-normalized derivation – especially in the context of label bias' influence on the brevity problem.

Conclusion

We have explored simple and effective ways to alleviate or eliminate the beam problem. We showed that the beam problem can largely be explained by the brevity problem, which results from the locally-normalized structure of the model. We compared two corrections to the model and introduced a method to learn the parameters of these corrections. Because this method is helpful and easy, we hope to see it included to make stronger baseline NMT systems.

We have argued that the brevity problem is an example of label bias, and that the solution is a very limited form of globally-normalized model. These can be seen as the simplest case of the more general problem of label bias and the more general solution of globally-normalized models for NMT BIBREF24 , BIBREF25 , BIBREF26 , BIBREF13 . Some questions for future research are:

Acknowledgements

This research was supported in part by University of Southern California, subcontract 67108176 under DARPA contract HR0011-15-C-0115, and an Amazon Research Award to Chiang.