Neural Machine Translation using Seq2Seq Attention Mechanism and various Decoding Strategies

Murali Mohana Krishna Dandu
UC San Diego (PID: A59004607)
mdandu@ucsd.edu

Abstract

This documents details the implementation results of a Seq2Seq Attention model for Machine Translation using the Multi30K dataset. Beam Search and Nucleus Sampling decoding strategies were implemented and compared with Greedy Search. The resulting output translations have been evaluated both qualitatively and quantitatively.

1 Introduction

Machine Translation is often modelled using sequence-to-sequence (seq2seq) RNN networks. The first architecture is the encoder RNN network which encodes the source sentence into a context vector. This vector can be often be thought of as an abstract representation of the entire input sentence. This vector is then used by the decoder to learn to output target sentence by generating one word at a time. There are a couple of downsides about this architecture. The context vector is only passed to the first hidden state and all the other hidden states will need to contain direct information from source sentence. Even though this problem can be solved by passing the context vector directly to RNN hidden and output states, another drawback is that the context vector still needs to contain all the information from the source sentence. To alleviate these, an attention mechanism is introduced by allowing the decoder to look at the entire source sentence at each step.

2 Approach

This section briefly discusses the network architectures and algorithmic details of Beam Search and Nucleus Sampling.

2.1 Seq2Seq RNN with Attention

The encoder network is bidirectional GRU layer where each hidden unit takes the input embeddings from the current time step and hidden state from the previous timestep. As the decoder is not bidirectional, the context vector z is created by concatenating the final hidden states of both RNN directions and applying a *tanh* activation.

$$h_t = EncoderGRU(e(x_t), h_{t-1})$$
$$z = tanh(g(h_{Tf}, h_{Tb}))$$

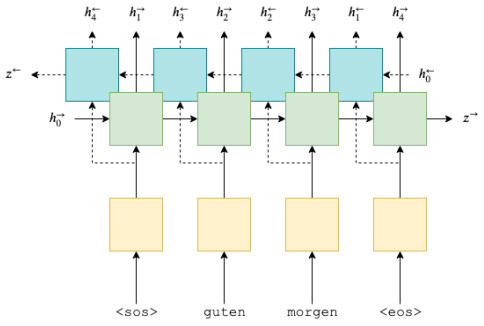


Figure 1: Encoder Architecture

The attention layer consists of energy mechanism followed by attention weights. The intuition behind energy mechansim is that we will use what all we have decoded so far along with the entire source sentence information. Next, the energy is transformed into an attention vector and passed through softmax layer to have weights between 0 and 1.

$$E_t = tanh(attention(s_{t-1}, H))$$
$$a_t = softmax(vE_t)$$

The first step of decoder is to calculate the weighted source vector using the previous attention weights and encoder hidden states. This additional input is passed into both the hidden state and the output linear layer. Also, the the target embedding is directly passed into the output later bypassing the hidden state.

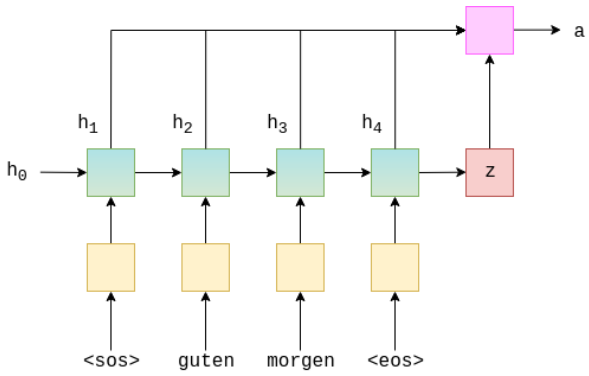


Figure 2: Attention Mechanism

$$w_t = a_t * H$$
$$s_t = DecoderGRU(d(y_t), w_t, s_{t-1})$$
$$\hat{y}_{t+1} = f(d(y_t), w_t, s_t)$$

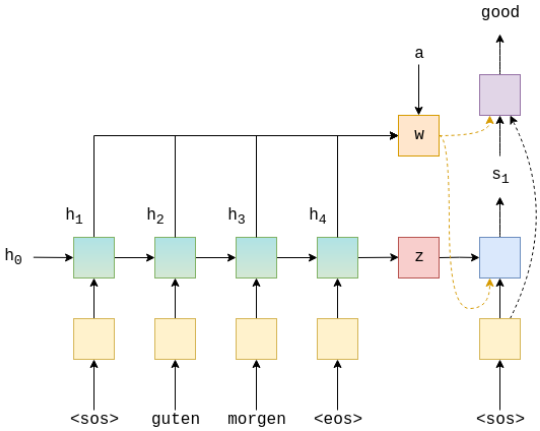


Figure 3: Decoder Architecture

2.2 Beam Search Decoding

Greedy decoding looks at maximum probability at each time step of output sequence which doesn't guarantee the complete sentence to have maximum likelihood. Beam Search addresses this issue by considering top k (beam width) conditional probability outputs at each time step.

- At a given time step, consider all the sequences generated by the previous top-k partial sequences
- Calculate their normalized log likelihood till that time-step and select top-k candidates
- If a sequence reaches <eos> token, don't generate new output but directly pass into our potential candidate list
- Repeat this process until all the top-k sentences reach <eos> token or we reach the maximum time step

In mathematical notation, we are trying to maximize the following:

$$argmax \frac{1}{T_y^\alpha} \sum_{t=1}^{T_y} logP(y^t|x, y^1, ...y^{t-1})$$

Couple of points to note regarding the above equation - To improve the numerical performance and avoid floating point vanishing, we use additives of log softmax while going through time steps. Also, to not simply favor short sequences , we normalize the scores at each time step by the sequence length and a softening parameter α.

2.3 Nucleus Sampling

Maximum likelihood decoding strategies sometimes cause output to degenerate - text that is repetitive and bland. To counter this and to produce variety, Nucleus Sampling is used for selective stochastic sampling. The core of this method lies in truncating the unreliable tail of probability distribution for each token, hence sampling from dynamic nucleus based on the how that particular sampling distribution looks like.

- First take the output probabilities at a given time step and identify the minimum number of tokens that form a top-p probability (nucleus).
- Re-normalize the distribution after making the rest of long-tail probs to zero
- Apply a multinomial sampling on the new distribution

Identifying smallest set of vocabulary that forms the p

$$\sum_x P(x|x_{1:i-1}) >= p$$

Re-normalizing the probability with the sum of the nucleus

$$P^{'}(x) = P(x)/p^{'}$$

This way, at any given time step, we are only considering such tokens that form the core (the core will be larger for flat distributions and smaller for skewed ones). I also varied the variety by using Temperature parameter on the logits before apply the above method.

3 Results and Discussion

3.1 Seq2Seq RNN with Attention

The above encoder-decoder RNN model with Attention is trained for 20 epochs using the following hyper-parameters: 256 embedding dim, 512 hidden layers and dropout probability of 50%. Fig. 4 shows that the training loss decreases and stabilizes, however the validation loss starts increasing after 10 epochs. This suggests that we may need to perform parameter tuning and regularization for improving the validation loss further. The train PPL reaches under 5 and the best validation PPL hits 25.

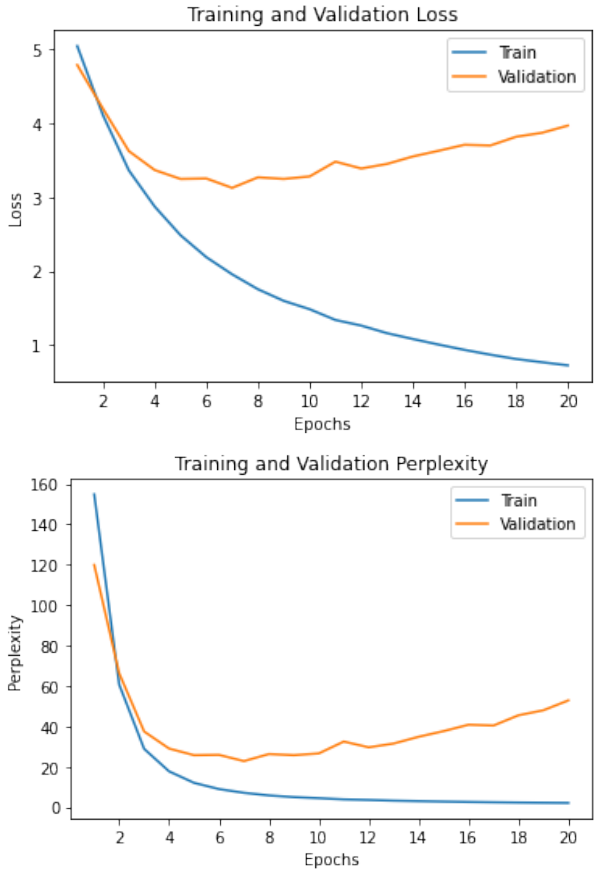


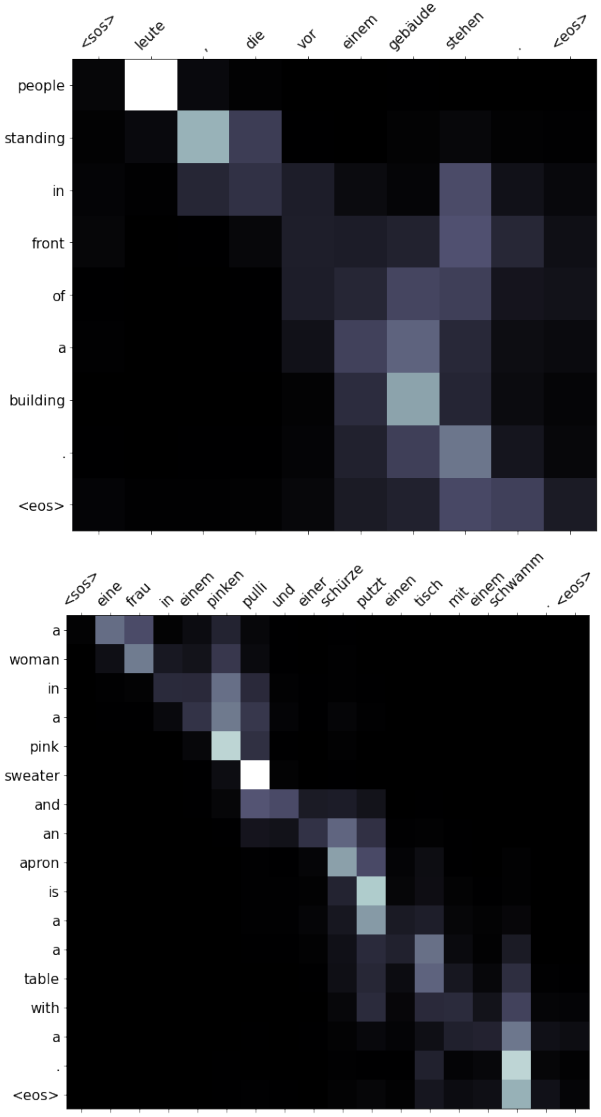
Figure 4: Loss and Perplexity Metrics

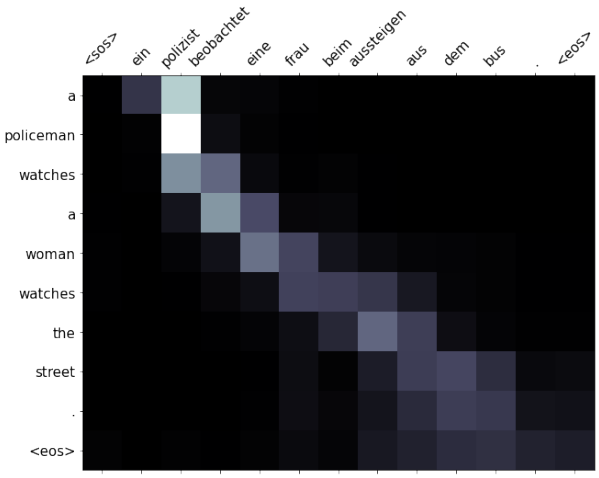
The best iteration test loss is **3.15** and Test PPL is **23.3**. I achieved a BLEU score of **28.88** on the test set which is in comparison with benchmark results.

Table 1 shows few translation samples from the above model. The attention vector heat-maps provide an idea on how much attention is given to the source words while translating that target word. I am also showing Google Translate as a reference.

Type	Sentence
SRC	leute , die vor einem gebäude stehen.
TRG	people standing **outside** of a building.
Google	people standing in front of a building.
Model	people standing in front of a building.
SRC	eine frau in einem pinken pulli und einer schürze putzt einen tisch mit einem schwamm.
TRG	a woman in a pink sweater and an apron , **cleaning a table with a sponge**.
Google	a woman in a pink sweater and apron is cleaning a table with a sponge.
Model	a woman in a pink sweater and an apron is a a table with a .
SRC	ein polizist beobachtet eine frau beim aussteigen aus dem bus .
TRG	police officer watching woman **exit from bus** .
Google	a policeman observes a woman getting off the bus.
Model	a policeman watches a woman watches the street

Table 1: Greedy Decoding test samples





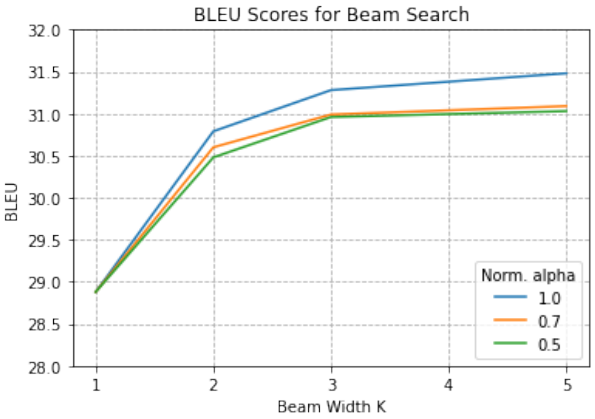


Figure 5: Beam Search BLEU Scores

In the first sample in Table 1, even though it isn't matching the exact target, it is able to provide a very clear translation matching Google's result. You can also see in the heatmap that the attention is able to rightly figure out the order of *'front of a building'* which is different to a word by word translation. In the other two samples which are slightly longer, model is not able to predict well in the second half of the sentence missing key noun phrases like *'cleaning .. sponge'*, *'exit from bus'*. This is also reflected in the right-bottom of the attention matrices where the attentions fade out. Below, I investigated if other decoding strategies improve these results.

3.2 Beam Search Decoding

The above mentioned beam search decoding algorithm has been implemented for multiple beams and normalization parameters. In Fig. 5, We can see that the BLEU score increases as we increase our beam width and k=5 gives a **BLEU score of 31.5 which is 2.5+ points higher** than the greedy search.

Clearly, we can see that the beam search is producing higher likelihood results and matching with human annotations. Also, we can see that the smoothening parameter is actually reducing the score when we move from 1 to 0.5. We observe that BLEU scores saturate at higher beams suggesting that larger beams won't be able to achieve similar improvement. Considering the algorithmic performance and saturation, k=5 (with no smoothening) seems like the ideal case for this particular set-up.

Table 2 shows output log likelihood across the entire test set compared across different beam widths. Note that the smoothening is applied for each sample and averaged for the whole test set, hence we cannot compare the likelihood across different smoothening values. The results clearly suggest that the higher beam widths are producing results which are more likely.

	$\alpha = 1$	$\alpha = 0.7$	$\alpha = 0.5$
k=1	-0.79	-1.78	-2.92
k=2	-0.74	-1.63	-2.75
k=3	-0.73	-1.60	-2.69
k=5	**-0.72**	**-1.58**	**-2.66**

Table 2: Beam Search Log Likelihood

Based on the examples in Table 3, we can see that the translation improves for certain samples at the end of the sentence matching exactly with the benchmarks. The phrases '*row of red chairs while*', '*holding hands in a pool*' are captured due to the exhaustive search that beam does compared to greedy. In the later two examples, even though the results don't exactly match with benchmarks, we still see that the translation make more sense compared to greedy.

3.3 Nucleus Sampling

Top-p nucleus sampling algorithm has been implemented with various top-p values and temperature parameters. Note that the existing BLUE score may not be the best metric to evaluate nucleus sampling results since the objective here is to sample for sentence variety. Also, note that since it is a stochastic algorithm, we get different variety of samples for the same nucleus size on different runs.

In Fig. 6, as we increase the nucleus value, the subset for selection increases which increases the diversity of the sample selection. This is the main reason that the BLEU score drops as we increase p value. If we are using p value around 0.1-0.2, we are limiting the sampling to a very top few words

Type	Sentence
SRC	eine frau sitzt in einer reihe aus roten stühlen und liest ein buch .
TRG	a woman reads a book while sitting in a **row of red chairs** .
Google	a woman is sitting in a row of red chairs reading a book.
Greedy	a woman is sitting in red , red , reading a book
Beam k=5	a woman is sitting in a row of red chairs while reading a book .
SRC	zwei mädchen in shorts halten händchen an einem pool .
TRG	two girls in shorts are **holding hands at a pool** .
Google	two girls in shorts hold hands by a pool.
Greedy	two girls in shorts holding goggles .
Beam k=5	two girls in shorts holding hands in a pool .
SRC	ein polizist beobachtet eine frau beim aussteigen aus dem bus .
TRG	police officer watching **woman exit from bus** .
Google	a policeman observes a woman getting off the bus.
Greedy	a policeman watches a woman watches the street
Beam k=5	a police officer watches a woman looks out of the bus .
SRC	eine gruppe von menschen auf einem obstmarkt im freien .
TRG	a group of people at an outdoor **fruit market**
Google	a group of people at an outdoor fruit market.
Greedy	a group of people are on an outdoor exercise . .
Beam k=5	a group of people on an outdoor fair . .

Table 3: Beam Search test samples

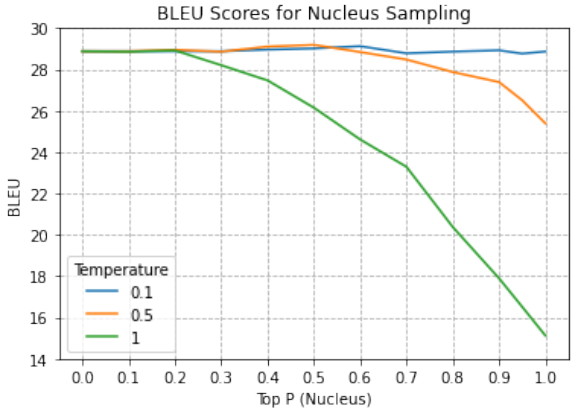


Figure 6: Nucleus Sampling BLEU Scores

which essentially behaves like the greedy search algorithm. Here, temperature is also used to skew the distribution while selecting the samples. For very small temperatures like 0.1, irrespective of the nucleus, the BLEU score remains consistent to that of greedy search. I further conducted a qualitative analysis using nucleus of 0.9 to compare diversity in samples keeping the temperature at 1.

Table 4 shows that the nucleus sampling provides a set of diversity to the translations mostly staying within the context of the source sentence. The first few examples have '*crew*', '*skateboarding over a pink top*', '*carnival outdoor festival*' which adds additional flavor to the context which is not achieved through greedy or beam search. However, the samples can go out of topic (based on the nucleus size as well) and may not well suited in several occasions. For example, the final samples

Type	Sentence
SRC	eine arbeitnehmerin macht eine pause , um etwas zu trinken .
TRG	an **employee** takes a break from working to have a drink
Google	an employee takes a break to have a drink.
Greedy	a unk taking a break break to take a drink .
Beam k=5	a unk takes taking break break to take a drink .
Nucleus p=0.9	a crew taking a break break to take to drink .
SRC	ein junger mann rutscht mit dem skateboard über ein rosa geländer .
TRG	a young man skateboards **off a pink railing** .
Google	a young man is sliding his skateboard over a pink railing.
Greedy	a young man slides down a a rail .
Beam k=5	a young man slides down a pink rail .
Nucleus p=0.9	a young man is skateboarding over a pink top above a skateboard .
SRC	eine gruppe von menschen auf einem obstmarkt im freien .
TRG	a group of people at an outdoor **fruit market**
Google	a group of people at an outdoor fruit market.
Greedy	a group of people are on an outdoor exercise .
Beam k=5	a group of people on an outdoor fair . .
Nucleus p=0.9	a group of people on carnival outdoor festival
SRC	ein hund springt durch ein brennendes hindernis .
TRG	a dog is **jumping** through a **fiery obstacle** .
Google	a dog is jumping through a burning obstacle.
Greedy	a dog jumps through an obstacle .
Beam k=5	a dog jumping through an obstacle course .
Nucleus p=0.9	a dog leaps through an indoor obstacle .
SRC	zwei mädchen in shorts halten händchen an einem pool .
TRG	two girls in shorts are **holding hands** at a pool .
Google	two girls in shorts hold hands by a pool.
Greedy	two girls in shorts holding goggles .
Beam k=5	two girls in shorts holding hands in a pool .
Nucleus p=0.9	two girls are wearing shorts holding guns while a pool .

Table 4: Nucleus Sampling test samples.

in Table 4 produce phrases '*indoor obstacle*', '*holding guns*' change the meaning and intent of these sentences. Hence, multiple samples with multiple nucleus have to be produced before selecting the suitable one.

4 Conclusion and Future Work

Seq2Seq Attention model has been implemented for Machine Translation and was analyzed both quantitatively and qualitatively. Beam search generated output samples with higher likelihood improving the BLEU score by 2.5 points. Computational time and improvement saturation has to be taken into consideration while selecting the best beam width. Nucleus sampling introduced some diversity to the samples by altering very few tokens. Given the randomness in its generation, it should be used in human-in-a-loop setting by checking different samples. Further, the model can be improved by using different attention mechanisms and architectures.