Project 1: Machine Translation

Nidhi Dhamnani
A59012902
CSE 291: Structured Prediction for NLP (Spring 2022)
E-mail: ndhamnani@ucsd.edu

***Abstract* – The goal of this project is to build a sequence-to-sequence (seq2seq) model with an attention mechanism for machine translation.**

***Keywords* – machine translation, seq2seq, beam search, nucleus sampling**

I. Dataset

In this project, we will be working on Multi30K machine translation dataset that is a large-scale dataset of images paired with sentences in English and German.

It consists of 1) German translations created by professional translators over a subset of the English descriptions and 2) German descriptions crowdsourced independently of the original English descriptions. To ensure an even distribution over description length, the English descriptions were chosen based on their relative length, with an equal number of longest, shortest, and median length source descriptions.

II. Seq2seq Model

A. *Training and Validation Loss*

The training loss indicates how well the model is fitting the training data, while the validation loss indicates how well the model fits new data.

As shown in Fig. 1, the training and validation loss decrease as we increase the number of epochs. We can see that the validation loss slightly increases as we move from epoch 8 to 10. The increase in validation loss along with decrease in training loss represents that the model has began to overfit.

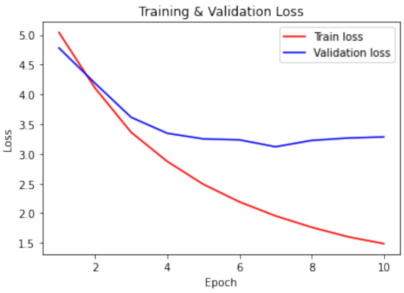

Fig. 1. Training and Validation Loss

B. *Attention Matrix*

Attention takes two sentences, one in each language, turns them into a matrix where the words of one sentence form the columns, and the words of another sentence form the rows, and then it makes matches. It helps to make connections between any particular word and its relevant context. The idea behind the attention mechanism is to permit the decoder to utilize the most relevant parts of the input sequence in a flexible manner.

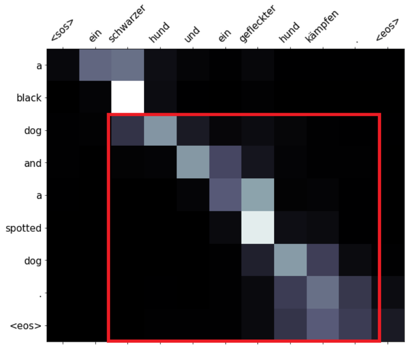

Fig. 2. Attention Matrix for Target Sentence: A black dog and a spotted dog are fighting

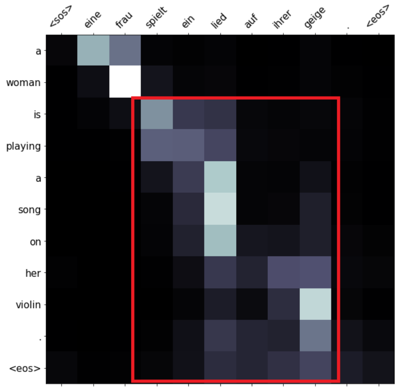

Fig. 3. Attention Matrix for Target Sentence: A female playing a song on her violin

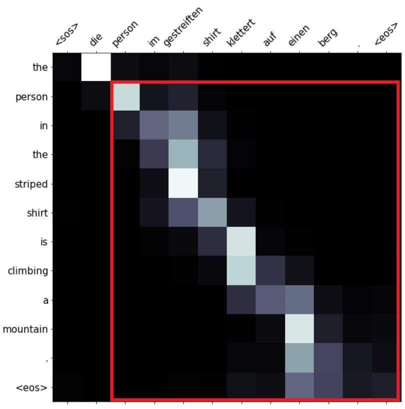

Fig. 4. Attention Matrix for Target Sentence: The person in the striped shirt is mountain climbing

The decoder takes a weighted combination of all of the encoded input vectors, with the most relevant vectors being attributed the highest weights.

In Fig 2 - 4, the source sentence on the x-axis and the predicted translation on the y-axis. The diagonal words are shaded with light colors whereas the non-diagonal words are shaded with dark colors. Here, the lighter the square at the intersection between two words, the more attention the model gave to that source word when translating that target word. In the areas marked as red, the model learned to swap the order of words in translation. We can also observe that this is not a 1-1 relationship but a many to many, meaning that an output word is affected by more than one input word, each one with different importance and vice versa.

C. *Translation Samples*

Greedy search is a simple translation algorithm. It selects the most probable word (i.e. argmax) from the model's vocabulary at each decoding time-step as the candidate to output sequence. Greedy decoding strategy is prone to have grammatical errors in the generated text.

Using table I-III, we can observe that it predicts most part of the sentence correctly. It replaces words at few places such as 'woman' instead of 'female' and 'climbing a mountain' instead of 'mountain climbing'. However, it is not changing the meaning of the original sentence, thus, preserving the semantics.

Type	Sentence
Source	ein schwarzer hund und ein gefleckter hund kämpfen .
Target	a black dog and a spotted dog are fighting .
Greedy Search	a black dog and a spotted dog .

TABLE I

Translation example on train data

Type	Sentence
Source	eine frau spielt ein lied auf ihrer geige .
Target	a female playing a song on her violin .
Greedy Search	a woman is playing a song on her violin .

TABLE II

Translation example on validation data

Type	Sentence
Source	die person im gestreiften shirt klettert auf einen berg .
Target	the person in the striped shirt is mountain climbing .
Greedy Search	the person in the striped shirt is climbing a mountain .

TABLE III

Translation example on test data

D. *BLEU Scores*

BLEU (BiLingual Evaluation Understudy) is a metric for automatically evaluating machine-translated text. It measures the similarity of the machine-translated text to a set of high quality reference translations. I obtained a score of 29.10 using the greedy search for decoding. Usually a BLEU score between 20-29 represents clear text with significant grammatical errors.

III. Beam Search

The beam search algorithm selects multiple alternatives for an input sequence at each timestamp based on conditional probability. The number of multiple alternatives depends on a parameter called beam width or beam size (k). Mathematically, it tries to maximize the following,

$$P(y|x) = P(y_1|x) * P(y_2|y_1,x) * \dots P(y_T|y_{T-1},\dots,y_1,x)$$

$$= \prod_{t=1}^{T} P(y_t|y_{t-1},\dots,y_1,x)$$

$$= \sum_{t=1}^{T} log(P(y_t|y_{t-1},\dots,y_1,x))$$

We search for a high scoring output sequence by keeping track of top-k vocabulary words at each timestamp while decoding.

When $k = 1$, it behaves like greedy search where *argmax* at any timestamp is fed to later step. When $k = size\ of\ vocabulary$, it behaves like an exhaustive search.

Beam search makes the following two improvement over greedy search:

- Instead of single best word, it expands and takes *k* best words at each step
- In greedy search, we pick each word in isolation without looking at previous words. Whereas in beam search we pick *k* best sequences so far by considers the probabilities of the combination of all of the preceding words along with the word in the current position.

A. *Implementation Details*

For implementing beam search, I followed the below steps:

- For each word prediction, maintained a list of all possible sequences called *all_candidates*
- Maintained the list of sequences which stores all the sequences of words seen so far
- If any sequence as reached $< eos >$, then directly add it to *all_candidates* without appending any other word to it. Else, go through the top-k words that have the highest log probabilities and append them to the current sequence. Finally, add the newly generated sequence to the list of all sequences, i.e. *all_candidates*
- Sort *all_candidates* in reverse order (increasing order of log probability) and store the top-k in the list of sequences
- Return the sequence with highest likelihood from sorted sequence list

B. *Beam vs Greedy Search Qualitatively Analysis*

From Table IV - VI, we can observe that neither beam search nor greedy search produce the exactly expected translation for all the three types of examples (train, val, and test). Both greedy and beam search perform similar for train example. For validation example, greedy performs better by detecting the color 'blue' of barrel, however, beam search predicts 'light' which is closer to word 'white' in terms of pronounciation but not close to 'blue', 'white', and 'barrel' in terms of meaning. For test example, beam search performs better by being able to detect the word 'holding' correctly. Greedy predicts same words 'in a kitchen' multiple times.

Type	Sentence
Source	ein schwarzer hund und ein gefleckter hund kämpfen .
Target	a black dog and a spotted dog are fighting .
Greedy Search	a black dog and a spotted dog .
Beam Search (k=7)	a black dog and a spotted dog .

TABLE IV
Translation example on train data

Type	Sentence
Source	drei kleine kinder stehen um ein blau-weißes fass herum .
Target	three young children stand around a blue and white barrel .
Greedy Search	three small children stand around a a blue vehicle .
Beam Search (k=7)	three small children stand around around a light .

TABLE V
Translation example on validation data

Type	Sentence
Source	eine frau , die in einer küche eine schale mit essen hält .'
Target	a woman holding a bowl of food in a kitchen .
Greedy Search	a woman in a kitchen in a kitchen in a kitchen .
Beam Search (k=7)	a woman holding a kitchen in a kitchen .

TABLE VI
Translation example on test data

C. *Beam vs Greedy Search Quantitatively Analysis*

The obtained BLEU score for greedy search and beam search with k=1 is same (= 29.01).

From Fig 5, we can see that the BLEU score increases as we increase beam size till a certain value (k=3) and then it starts to decrease. The graph is a deviation from the expected result where we expect the BLEU score to increase as we increase beam size. This happens probably because as the beam size increases, the number of candidates to be explored increases. Therefore, it becomes easier for the algorithm to find $< eos >$ token. Also, as beam size increases, the search algorithm generates shorter candidates, and then prefer even shorter ones among them.

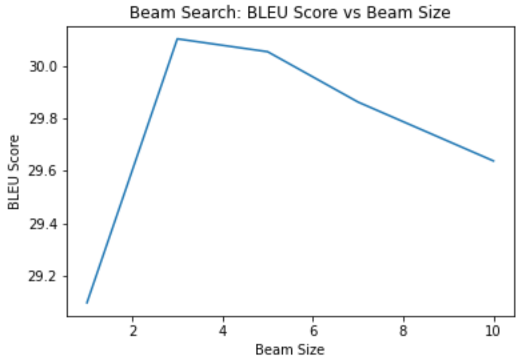

Fig. 5. Beam Search Performance

From Table VII-VIII, we can also observe that the sentence with the highest likelihood need not necessarily give the best prediction. In Table VII, the second highest likelihood result (with a score of -6.57) is indeed closer to the expected result and best amongst the top-5 sentences.

Likelihood	Sentence
-6.28	three small children stand around around a light .
-6.57	three small children stand around around a blue vehicle .
-6.66	three young children stand around around a light .
-6.73	three small children stand around around a light . .
-6.79	three small children are around around a light .

TABLE VII
Translation Likelihood for Top-k Sentences on Validation Data (k=5)

Likelihood	Sentence
-9.00	a woman holding a kitchen in a kitchen .
-9.06	a woman in in a kitchen in a kitchen .
-9.57	a woman in a kitchen in a kitchen with food .
-9.74	a woman holding a in in a kitchen in a kitchen .
-9.87	a woman in a kitchen in a kitchen in a kitchen .

TABLE VIII
Translation Likelihood for Top-k Sentences on Test Data (k=5)

IV. Nucleus Sampling

In nucleus sampling, we focus on selecting the smallest possible sets of Top-V words such that the sum of their probability is $\geq$ p (some probability value). Then, the probability of tokens that are not in Top-V are set to 0 and the rest are re-scaled to ensure that they sum to 1. Formally,

$$\sum_{x \in V^{(p)}} P(x|x_{1:i-1}) \geq p$$

A. *Implementation Details*

For implementing nucleus sampling, I followed the below steps:

- Convert the logits to probability scores using *softmax* function
- Sort the tokens based on probability scores and also store the indicies of the original tokens
- Calculate the cumulative probability of the sorted tokens
- Remove the tokens where cumulative probability is less than p
- Redistributed / normalize the probability amongst the remaining tokens
- Apply multinomial distribution on the normalized tokens and randomly select the token from the distribution
- In case all tokens for a given row have cumulative probability $< p$, select the token using greedy search

B. *Nucleus vs Greedy Search Qualitatively Analysis*

From Table IX - XI, we can observe that neither nucleus search nor greedy search produce the exactly expected translation for all the three types of examples (train, val, and test). Both greedy and nucleus search perform similar for train and test examples. For validation example, both fail to predict the words 'and white barrel' and predict incorrect/unrelated words.

Type	Sentence
Source	ein schwarzer hund und ein gefleckter hund kämpfen .
Target	a black dog and a spotted dog are fighting .
Greedy Search	a black dog and a spotted dog .
Nucleus Search (p=0.4)	a black dog and a spotted dog .

TABLE IX
Translation example on train data

Type	Sentence
Source	drei kleine kinder stehen um ein blau-weißes fass herum .
Target	three young children stand around a blue and white barrel .
Greedy Search	three small children stand around a a blue vehicle .
Nucleus Search (p=0.4)	three small children stand around a a blue cloth .

TABLE X
Translation example on validation data

Type	Sentence
Source	eine frau , die in einer küche eine schale mit essen hält .'
Target	a woman holding a bowl of food in a kitchen .
Greedy Search	a woman in a kitchen in a kitchen in a kitchen .
Nucleus Search (p=0.4)	a woman in a kitchen in a kitchen in a kitchen .

TABLE XI
Translation example on test data

C. *Nucleus vs Greedy Search Quantitatively Analysis*

From Fig. 6 we can observe the effect of thresold probability p on the BLEU score. We can observe that when p is less (ex: p = 0.4), the obtained BLEU score is closer to the greedy search and it keeps on decreasing as we increase p. As we increase p, the number of words from which we sample increase. For our dataset, this results in increase in non-relevant token in the sample space. Therefore, when we sample a token from the set of the tokens shortlisted by $top - V^{(p)}$, the chances of getting gibberish words increase and hence the BLEU score decreases.

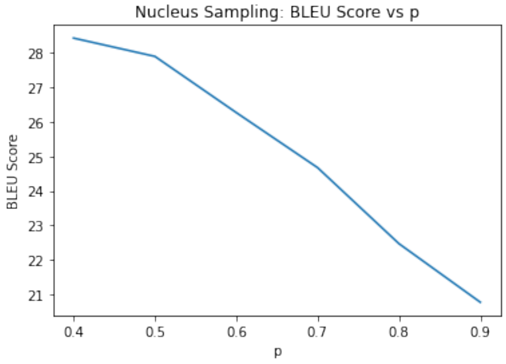

Fig. 6. Nucleus Sampling Performance

V. Conclusion: Comparing Greedy, Beam, and Nucleus Search

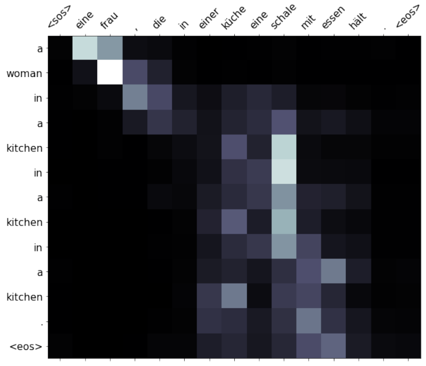

Fig. 7. Test Data: Greedy Search Attention Example

From the Fig. 7-9, we can observe that the sentences generated by greedy and nucleus sampling are larger than the sample generated by beam search. In all the three cases, there is a lot of ambiguity and rearrangement of words for later part of the source sentence. We cannot comment which sampling is better from the generated attention matrices.

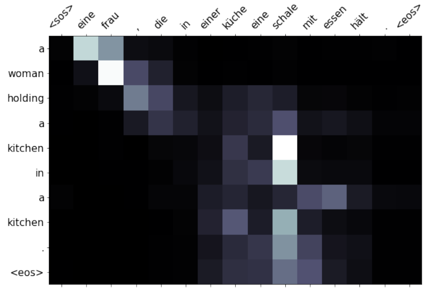

Fig. 8. Test Data: Beam Search Attention Example

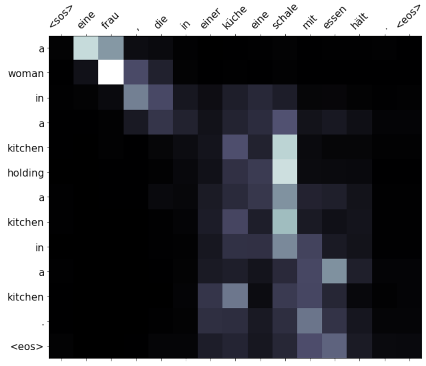

Fig. 9. Test Data: Nucleus Search Attention Example

From the BLEU scores, beam search seem to perform the best. However, from the translated sentences, we can observe the apparent flaw in all the three strategies - generation of repetitive words and meaningless sentences. Finally, one cannot clearly comment on which decoding technique is the best and the result depends on the evaluation criteria and use case.

VI. References

1. Multi30K: Multilingual English-German Image Descriptions
2. What is a Learning Curve in Machine Learning?
3. A Beginner Guide to Attention
4. How Attention works in Deep Learning
5. The Attention Mechanism from Scratch
6. Evaluating Models
7. Word Sequence Decoding in Seq2Seq Architectures
8. Nucleus Sampling for Natural Language Processing
9. Foundations of NLP Explained Visually: Beam Search, How It Works
10. An intuitive explanation of Beam Search
11. The Curious Case of Neural Text Degeneration
12. Decoding Strategies that You Need to Know for Response Generation
13. Breaking the Beam Search Curse