1. Group Information: Group Name: entarter; **Group Members:** Sihui Wang, Mohammad Dorkhah

2. Task: German-to-English Machine Translation with Attention **Input:** German text to be translated **Output:** the translated English text **Goal:** Improve BLEU scores (which is a mild indicator of translation quality)

3. Methods:

3.1 Baseline Method: In the default solution, we already have the encoder-decoder architecture for neural machine translation. However, in the default solution, the attention mechanism between the encoder and the decoder is not properly implemented. With poorly predicted context information, the network is unable to make good translations. In baseline method, we correctly implemented attention mechanism between the encoder and the decoder.

During each time step t in machine translation, we use the decoder's hidden state, h^{dec}, and the encoder's outputs, h_i^{enc} (which corresponds to the i-th word in the source text), to compute the attention scores for the i-th word:

$$score_i = W_{enc}(h_i^{enc}) + W_{dec}(h^{dec})$$

$score_i$ describes the relevance between the i-th word in the source text and the current target word to be translated.

Then, we use the *softmax* function to normalize the scores, so that they sum to 1:

$$\alpha = softmax(V_{att} \cdot \tanh(score))$$

Here, α is the vector $\alpha = (\alpha_1, \dots, \alpha_n)$ of attention weights. Each α_i describes how the original word at position i (encoded as h_i^{enc}) is relevant to the next word to be translated.

Then, we use the attention weights to calculate the context vector c, which is a summary of the most relevant parts in the source text for the target:

$$c = \Sigma_i \alpha_i h_i^{enc}$$

With the context vector c, the decoder can get access to the summary of relevant information from the source during translation, which is helpful for improvement of translation quality.

3.2 Beam Search Decoding: At each time step t, the decoder generates the probabilistic distribution of the next word. Yet we still need to find an optimal sequence of text given the probabilistic distributions. In method 3.1, the decoder always seeks the locally optimal solution (greedy decoding). In method 3.2, we try to improve the decoding technique so that the decoder can generate a sequence with better global optimality.

Suppose that at time step t, given the last translated word x_t, encoder's output h^{enc}, and decoder's hidden state h_t^{dec} at time t, the decoder returns:

$$\tilde{x}_{t+1}, h_{t+1}^{dec}, \alpha_t = Decoder(x_t, h^{enc}, h_t^{dec})$$

Here $\tilde{x}_{t+1}$ is a distribution of the probability of the next predicted word. In greedy decoding, we simply choose the word which maximize this probability as the next word x_{t+1}:

$$x_{t+1} = \arg\max_{v \in V} \tilde{x}_{t+1}(v)$$

To some degree, beam search decoding is a generalization of the greedy decoding. Instead of choosing only 1 candidate word, x_t, at each time step t, in beam search decoding, we keep the top k words, $x_{t,1}, \dots, x_{t,k}$, at each time step t.

In beam search, at each time step t, we keep the k most probable words: $x_{t,1}, \dots, x_{t,k}$. For each candidate word $x_{t,i}$, we calculate:

$$\tilde{x}_{t+1,i}, h_{t+1}^{dec}, \alpha_t = Decoder\left(x_{t,i}, h^{enc}, h_t^{dec}\right)$$

Here $\tilde{x}_{t+1,i}$ is a distribution of the probability of the next predicted word, given that $x_{t,i}$ is the word at time step t.

For each distribution $\tilde{x}_{t+1,i}$, we can find the top k candidate words, $x_{t+1,i,1}, \dots, x_{t+1,i,k}$:

$$x_{t+1,i,1} = \arg\max_{v \in V} \tilde{x}_{t+1,i}(v)$$

$$x_{t+1,i,2} = \arg\max_{v \in V - \{x_{t+1,i,1}\}} \tilde{x}_{t+1,i}(v)$$

……

$$x_{t+1,i,1} = \arg\max_{v \in V - \{x_{t+1,i,1}, \dots, x_{t+1,i,k-1}\}} \tilde{x}_{t+1,i}(v)$$

So, for time step $t+1$, we find k^2 candidate words, $x_{t+1,i,j}$ $(1 \le i \le k, 1 \le j \le k)$. From $x_{t+1,i,j}$ $(1 \le i \le k, 1 \le j \le k)$ we choose the k most probable words as the k candidate words $x_{t+1,1}, \dots, x_{t+1,k}$ for the time step $t+1$.

Since $x_{t+1,1}, \dots, x_{t+1,k}$ are generated from $x_{t+1,i,j}$ $(1 \le i \le k, 1 \le j \le k)$, and $x_{t+1,i,j}$ $(1 \le i \le k, 1 \le j \le k)$ are generated from $x_{t,i}$, we can trace back which $x_{t,i}$ generates the k candidate words at time step $t+1$. Hence, at each time step t, we can find the k *so far the most probable* sequences.

We keep generating the k most probable words at each time steps, until we finish the whole sentence. Then, in the k most probable sequences, we pick the most probable one, which completes beam search and generates our final translation.

3.3 Post Processing: Replication Removal The translation generated by method 3.1 and method 3.2 contains repetitive words. Such repetitive words can negatively affect the BLEU score, because in the reference translation there is no n-gram term with repetitive words. In method 3.3, we proposed to examine the outputs of the decoder and remove such repetitive words to improve the BLEU scores.

What we did for replication removal can be summarized as follows:

For the sentence with the sequence of words: $w_1, w_2, \dots, w_n$, keep w_i $(2 \le i \le n)$ in the post-processed sentence *if and only if* $w_i \neq w_{i-1}$.

3.4 Post Processing: <UNK> Replacement After the post processing from method 3.3, we still have <UNK> tokens in the translation outputs. Since in the reference translation there is no n-gram term with <UNK> tokens, removing <UNK>, or replacing <UNK> with any frequent words in English, would improve BLEU scores. However, such methods might lead to worsened translation quality.

In our work, we proposed to replace <UNK> with its corresponding word or words in the original text.

In the attention mechanism, for each target word, we have computed the alpha vector, which is the target word's relevance to every source words. Now, for the whole sentence with target words $w_1, \dots, w_n$, we obtain n alpha vectors, $\alpha_1, \dots, \alpha_n$. We can organize all $\alpha_1, \dots, \alpha_n$'s parameters in a matrix $A = (a_{ij})$ so that a_{ij} describes the relevance between the source word i and the target word j.

Then, for each source word i, we compute p_i, which describes which target word the i-th source word is the most relevant to:

$$p_i = \arg\max_j \alpha_{ij}$$

When we encounter an <UNK> token at position idx in the target text, we check all p_i to see if there is any $p_i = idx$. If so, then the i-th source word is the most relevant to the idx-th target word and we replace the <UNK> token in the idx-th position by the i-th word in the original text.

3.5 "Ensemble" of Models

Note: In this part we are not implementing *ensemble decoding*.

In method 3.5, our goal is to use a linear interpolation of different model's outputs for generation of the translated text.

Assume that we have M models, and each of the M models have generated a decoded sequence O_i $(1 \leq i \leq M)$ and a sequence of attention vectors $A_i (1 \leq i \leq M)$. What we are doing is to calculate the linear combination of O_is and A_is:

$$O = \Sigma_{i=1}^{M} \lambda_i O_i, A = \Sigma_{i=1}^{M} \lambda_i A_i$$

Here, $\lambda_i (1 \leq i \leq M)$ is the weights for the model i. Instead of using O_i and A_i from a single model during translation, we use the linear interpolation O and A, where O predicts the target words at each location, and A predicts which part of the source text is relevant to which target word. Since we used the linear interpolation of model predictions to guide the translation process, we believe that our approach adopts "ensemble" of models.

4. Results:

In our work, we did experiments for the following methods:

The default method (def): this is an encoder-decoder architecture with no properly implemented attention mechanism.

The baseline method (bs): this is the same encoder-decoder architecture with properly implemented attention mechanism.

The beam search method (bm): bm adopts the same encoder-decoder architecture and attention mechanism as in bs. The difference is that in bm, beam search decoding is implemented in the decoder.

Beam search with replication removal (bm+rr): A post-processing technique called replication removal is added to bm method.

Beam search with replication removal and <UNK> replacement (bm+rr+ur): Another post-processing technique called <UNK> replacement is added to the bm+rr method.

Ensemble of models for translation (bm+rr+ur+en): We use the same network architecture, attention mechanism, and post-processing techniques as in bm+rr+ur. In the translation process, instead of relying on one model's prediction, in bm+rr+ur+en, we use a linear interpolation of different models' predictions to guide the translation process.

4.1 Quantitative Results on Dev Dataset:

We tested the above-mentioned methods on Dev dataset. We evaluate different methods' translation quality by BLEU scores, and below are the quantitative results:

Note: for the methods which adopt beam search, we also report the additional parameter, the beam width k, that we used in the experiments;

For non-ensemble methods, we used the model "seq2seq_E049.pt"; for the ensemble method, we used the linear interpolation of 5 models: "seq2seq_E045.pt", "seq2seq_E046.pt", "seq2seq_E047.pt", "seq2seq_E048.pt", and "seq2seq_E049.pt".

Quantitative Comparison of BLEU Score Among Different Methods

	def	**bs**	**bm ($k = 5$)**	**bm+rr ($k = 5$)**	**bm+rr+ur ($k = 5$)**	**bm+rr+ur+en ($k = 5$)**	**bm+rr+ur+en ($k = 25$)**
BLEU score	3.3529	17.1139	19.0377	19.3868	19.7125	19.2279	20.3072

From the quantitative results we can see that attentional mechanism, replication removal, and <UNK> replacement all contributes positively to BLEU scores. However, comparing with bm+rr+ur ($k = 5$), our "ensemble" method bm+rr+ur+en ($k = 5$) actually leads to lower BLEU score.

We also compared how different beam search widths affect the BLEU scores.

Quantitative Comparison of BLEU Score Among Different Beam Search Width

	bm+rr+ur ($k = 5$)	**bm+rr+ur ($k = 10$)**	**bm+rr+ur ($k = 15$)**	**bm+rr+ur ($k = 20$)**	**bm+rr+ur ($k = 25$)**	**bm+rr+ur ($k = 30$)**	**bm+rr+ur ($k = 35$)**
BLEU score	19.7125	19.9215	20.1964	20.2215	20.3072	20.2629	20.3003

From the above results we can see that, initially, increasing beam search width can significantly improve the BLEU score. However, in our experiments, after the beam search width reached $k = 25$, further increasing the beam search width won't guarantee an improvement in BLEU scores.

4.2 Qualitative Results:

Ground Truth

Source Text (German)	**Translated Text (English)**
Als ich in meinen 20ern war , hatte ich meine erste Psychotherapie-Patientin .	When I was in my 20s , I saw my very first psychotherapy client .

Translation Results by Different Methods

Here are the translation results generated by different methods.

Method	**Translated Text (English)**
def	i was , i had my my , i had my .
bs	when when i was in my 20s , i had my first <unk> . .
bm ($k = 5$)	when when i was in my 20s , i had my first <unk> . .
bm+rr ($k = 5$)	when i was in my 20s , i had my first <unk> .
bm+rr+ur ($k = 5$)	when i was in my 20s , i had my first psychotherapie-patientin .
bm+rr+ur+en ($k = 5$)	when i was in my 20s , i had my first psychotherapie-patientin .
bm+rr+ur+en ($k = 25$)	when i was in my 20s , i had my first psychotherapie-patientin .

5. Discussion and Analysis:

5.1 Attention is Crucial for Neural Machine Translation:

From both qualitative and quantitative results we can confirm that attention mechanism greatly improves the translation quality measured either by BLEU scores or human evaluation. Without attention mechanism, the default method can still make correct translations for some of the words in the sentence (for example, the default method correctly translated "I was" in the sentence "When I was in my 20s …", and it correctly translated "she was a woman" in the sentence "She was a 26-year-old woman named Alex"). However, without attention, the default method seems to only be able to recognize some of the most frequent words or phrases in English, and it makes a lot of repetitive translations. The rich information in the sentence, is not attended to by the model and lost. From such results we can conclude that, without attention, the encoder's last hidden state is a bottleneck of information, which can't provide a good summary of the source sentence. With attention mechanism, the model can attend to the most relevant part in the source text for the translation of each target words, which greatly improves the quality of neural machine translation.

5.2 Larger Beam Search Width is Not Always Good

From the quantitative results we can see that the BLEU score peaks when the beam width is around $k = 25$. After the beam search width reaches $k = 25$, further increasing the beam search width helps little for improving BLEU scores. In fact, it is a little bit counter-intuitive that increasing k from 25 to 30 actually leads to lower BLEU scores. Our preliminary explanation for this is that, while increase the beam search width ensures that the algorithm can search over a larger space to find the "optimal" solution, the "optimality" defined by the training dataset might be different from the "optimality" required in the testing set. As we increase the beam search width, the algorithm makes a more

thorough search, and the solution might converge to the optimality according to the training set, however, this doesn't guarantee that the solution is optimal for the testing dataset.

5.3 Translation Quality is Different From BLEU Scores

If the translated text contains some tokens or patterns of tokens which would never appear in the reference translation text, then such tokens or patterns of tokens would negatively affect the BLEU score. Removing such tokens, or replacing such tokens with some frequent words in English, can always be helpful for improving the BLEU score. However, such technique might actually leads to worsened translation quality, because the text can become confusing after removal of tokens or replacement of the tokens with some random words. What we learned from the experiments is that we still want to replace such tokens with some relevant information, so that we can preserve the information in the text, and improve the BLEU score without sacrificing the translation quality.

5.4 Why Ensemble Fails?

In our experiments, we tested the "ensemble" of models, however, the result of ensemble is not as good as the same method without "ensemble". We have 2 preliminary explanations for this: 1) our implementation of "ensemble" is not good, and its result is still far from the results that we can get from ensemble decoding. 2) we guess that we need to use a number of fundamentally different models, so that the ensemble of models can produce a good results. In our experiments, all models are using the same network architecture, and we guess that their only possible differences are the training epochs and parameters. Since all models use the same network architecture, they share the same advantages and drawbacks. Since their only difference might be the training epochs, maybe in the 5 models, some are just better than others because they are trained with more epochs. We guess that in ensemble learning, we actually want the models to have different advantages, so that they can cover each other's drawbacks. In our experiments, all 5 models are "homogenous" with the same network architecture, same advantages, and same drawbacks, which might be a reason why ensemble of such models won't produce good results.

6. Contributions:

Sihui Wang implemented the baseline method, beam search decoding, the post processing techniques of replication removal and <UNK> replacement, and the "ensemble" of models in the translation process.

Sihui Wang did experiments with different parameter settings to find the best-so-far result.

Sihui Wang made documentations for the code.

Sihui Wang performed analysis and completed the write-up of the notebook and the report.

Mohammad implemented the baseline method.