CS626
COURSE PROJECT

EFFICIENT NEURAL MACHINE TRANSLATION

ANSHUL TOMAR, 170070007
ANWESH MOHANTY, 170070009
PARTH SHETTIWAR, 170070021

Neural Machine Translation (NMT)

- Machine Translation (MT) is the task of automatically converting of written text from one natural language to another, while also preserving the meaning of the input text.
- Neural Machine Translation models have replaced traditional count based Statistical Machine Translation models.
- NMT models currently give state-of-the-art (SOTA) performances in almost all of the machine translation related tasks.

But there are still severals issues faced by NMT models.

Problems Encountered in NMT models

1. Overfitting on training dataset: Occurs primarily due to high complexity of RNN models and due to discrete nature of training inputs.

   This leads to fall in results during the inference stage.

2. Training time: Current SOTA models take around 2-3 days to train properly on large datasets to produce their results. But there might be cases in which such large training times might not be desirable.

```
Epoch: 01 | Time: 3m 9s
        Train Loss: 3.730 | Train PPL:  41.687
         Val. Loss: 3.395 |  Val. PPL:  29.818
Epoch: 02 | Time: 3m 10s
        Train Loss: 2.639 | Train PPL:  13.993
         Val. Loss: 3.135 |  Val. PPL:  22.978
Epoch: 03 | Time: 3m 10s
        Train Loss: 2.215 | Train PPL:   9.164
         Val. Loss: 3.165 |  Val. PPL:  23.680
Epoch: 04 | Time: 3m 9s
        Train Loss: 1.952 | Train PPL:   7.041
         Val. Loss: 3.216 |  Val. PPL:  24.921
Epoch: 05 | Time: 3m 11s
        Train Loss: 1.788 | Train PPL:   5.979
         Val. Loss: 3.342 |  Val. PPL:  28.267
Epoch: 06 | Time: 3m 10s
        Train Loss: 1.681 | Train PPL:   5.370
         Val. Loss: 3.458 |  Val. PPL:  31.740
Epoch: 07 | Time: 3m 9s
        Train Loss: 1.619 | Train PPL:   5.047
         Val. Loss: 3.561 |  Val. PPL:  35.215
Epoch: 08 | Time: 3m 10s
        Train Loss: 1.577 | Train PPL:   4.841
         Val. Loss: 3.545 |  Val. PPL:  34.624
Epoch: 09 | Time: 3m 9s
        Train Loss: 1.550 | Train PPL:   4.712
         Val. Loss: 3.627 |  Val. PPL:  37.591
Epoch: 10 | Time: 3m 10s
        Train Loss: 1.547 | Train PPL:   4.699
         Val. Loss: 3.743 |  Val. PPL:  42.212
```

Overfitting in Baseline Model

Our Contributions

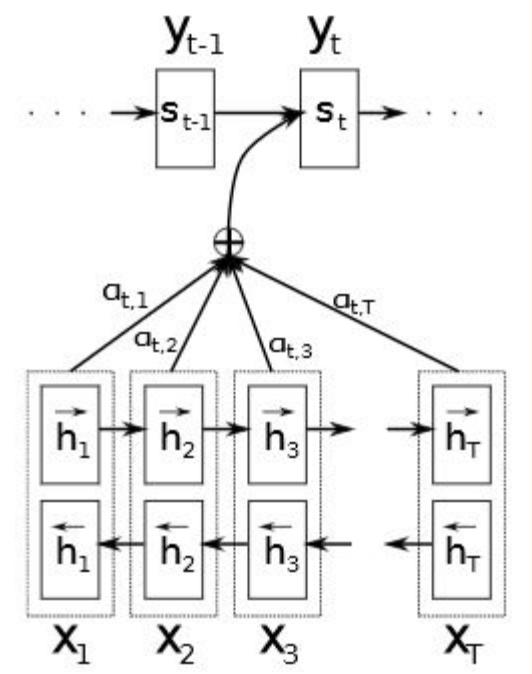
- Built a NMT model based on the popular RNNSearch model but with a smaller memory footprint and lower computation cost compared to SOTA models but accuracy isn't compromised much.
- Added an adversarial training mechanism to our model to tackle overfitting to some extent.
- Presented a detailed characterization of model performance and how each component affects it.
- Conducted a thorough comparative study with a SOTA transformer to show how effectively we have achieved our goals.

Model Architecture

Our model has three components:

1. Encoder: Consists of an Embedding layer, bi-directional GRU and a Linear layer
2. Attention Layer: A feedforward neural network to implement attention
3. Decoder: Consists of an Embedding layer, GRU and a Linear layer

Additionally later on in our experiments we also implement the adversarial training experiments in the encoder and decoder.



Adversarial MLE Training

- Expt1: We add random noise to hidden states of the encoder.
- Expt2: We implement the adversarial MLE training algorithm to the decoder as shown here.

Both the experiments try to stop overfitting by ensuring that variations are introduced in the hidden states and the model remains generalized.

Adversarial MLE training also constrains the hidden state representations such that they are spaced from each other in the latent space and do not interfere with each other while making predictions due to close proximity of hidden state representations.

Algorithm 1 Adversarial MLE Training

Input Training data $\mathcal{D} = \{x^{\ell}_{1:T}\}$, model parameters θ, w

while not converge **do**

Sample a mini-batch $\mathcal{M}$ from the data $\mathcal{D}$.

For each sentence $x^{\ell}_{1:T}$ in the minibatch and $t \leq T$, set the adversarial noise on $p(x^{\ell}_t | x^{\ell}_{1:t-1})$ to be

$$\delta_{j;t,\ell} = \begin{cases} -\epsilon h^{\ell}_t / ||h^{\ell}_t||, & \text{for } j = x^{\ell}_t \\ 0, & \text{for } j \neq x^{\ell}_t, \end{cases}$$

where h^{ℓ}_t is the RNN hidden state related to $x^{\ell}_{1:t-1}$, define in (2).

Update $\{\theta, w\}$ using gradient ascent of log-likelihood (4) on minibatch $\mathcal{M}$,

end while

Dataset: Multi30K German to English

- Originally planned to use WMT14 De to En but faced resource constraints
- Statistics of Multi30K dataset:
  - Training:
    - En: 29000 sentences, 377534 words, 13.0 words/sent
    - De: 29000 sentences, 360706 words, 12.4 words/sent
  - Validation:
    - 1014 sentences, 13308 words, 13.1 words/sent
    - 1014 sentences, 12828 words, 12.7 words/sent
  - Test:
    - 1000 sentences, 12968 words, 13.0 words/sent
    - 1000 sentences, 12103 words, 12.1 words/sent

Results and Ablation Study

Translation of sentences with epoch

We analysed the translation of sentences after every epoch and analysed how the related words show up in translations

Input Sentence: Eine junge Dame macht Yoga am Strand .

Epoch 1: a man in a a a .
Epoch 2: a man is a a a a .
Epoch3: a young woman is a a a .
Epoch 4: a young woman is a a the ocean .
Epoch 5: a young lady is a on the beach .
Epoch 6: a young lady is on the beach on the beach .
Epoch 7: a young lady is swinging on the beach .
Epoch 8: a young lady is on the on the beach .
Epoch 9: a young lady is doing yoga on the beach .

We see that related words like ocean and man(related to beach and lady) show up in starting epochs, but our network learns the correct translation eventually.

Comparison with Transformer

	Our Model	Transformer
Parameters	8.8 Million	25.96 Million
Time per Epoch	28 sec	40 sec
Val. Perplexity(PPL)	27.2	5.773
BLEU Score	31.34	34.25

- We have used the Transformer as described in paper “Attention is All you Need” by Vaswani et.al
- We observe that Transformer takes much more time and parameters to achieve a BLEU score of 34.25. Whereas our model takes about 3 times less parameters and about 1.4 times less time per epoch to achieve a decent Bleu score of 31.34.

Error Analysis

For the error analysis of our model, we analysed the translated sentences of our test dataset and compared them with groundtruth.

- Firstly, many a times our translation gives repeated words as translation. A striking example for this is:

  Input Sentence: *Zwei Männer reiten über eine Farm und führen einen von einem Esel gezogenen Wagen .*

  Translated Sentence: **Two men are riding a a a a a of a donkey .**

  Groundtruth Sentence: **Two men ride across a farm and lead a wagon pulled by a donkey.**

  We speculate that there is some collapse happening which triggers a chain reaction and our model outputs same word many times. Or it might be possible that we didn't run the model for enough epochs.

- Sometimes it happens that even though the word is present in the vocabulary (generated from training data), the translated sentence shows out of vocabulary word (we have used tag <unk>) or some other rare word. This happens specifically when the number of times the word has appeared in training data is very less. For example:

  Input Sentence: *Ein Reh springt über einen Zaun .*

  Translated Sentence: A free athlete is jumping over a fence .

  Ground Truth Sentence: A deer jumps a fence .

  In this example, the input sentence word “Reh” (which means deer in english) has come only 3 times in our training data. For such cases we have observed wither erroneous predictions or <unk> being predicted in the final translated sentence.

- It was also observed that, our model interchangeably gives the continuous form and 3rd Person Singular form of Verb while translating the sentence. For example:

  Input Sentence: *Ein kleines Mädchen hält einen kleinen Jungen auf ihrem Schoß .*

  Translated Sentence: A little girl **holds** a little boy on her lap .

  Ground Truth Sentence: A little girl **is holding** a little boy on her lap .

  We speculate this may be due to the inaccuracies in the training set. Multi30k is a manually translated dataset, for which it might happen that these 2 forms of verb are interchangeably used while preparing dataset.

BLEU Score vs Length of Sentence

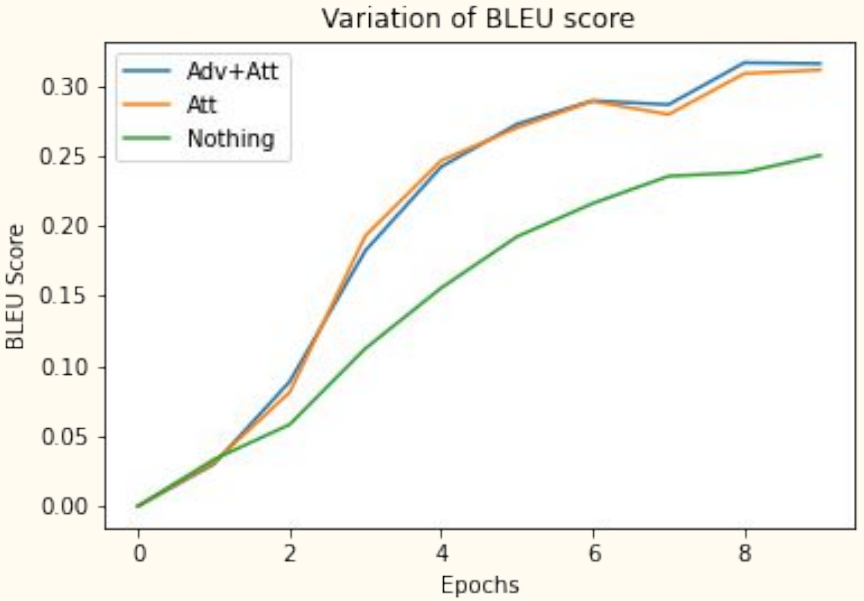
Here we calculate the performance of our model on different range of lengths of sentences and calculate the BLEU scores on them. This will help us better identify sentences on which the model performs poorly.

Length of Sentence (L)	Number of sentences in test dataset	BLEU Score
$L \leq 5$	7	19.64
$6 \leq L \leq 10$	390	33.33
$11 \leq L \leq 15$	432	32.40
$16 \leq L \leq 25$	162	28.32
$26 \leq L$	14	23.54

Ablation

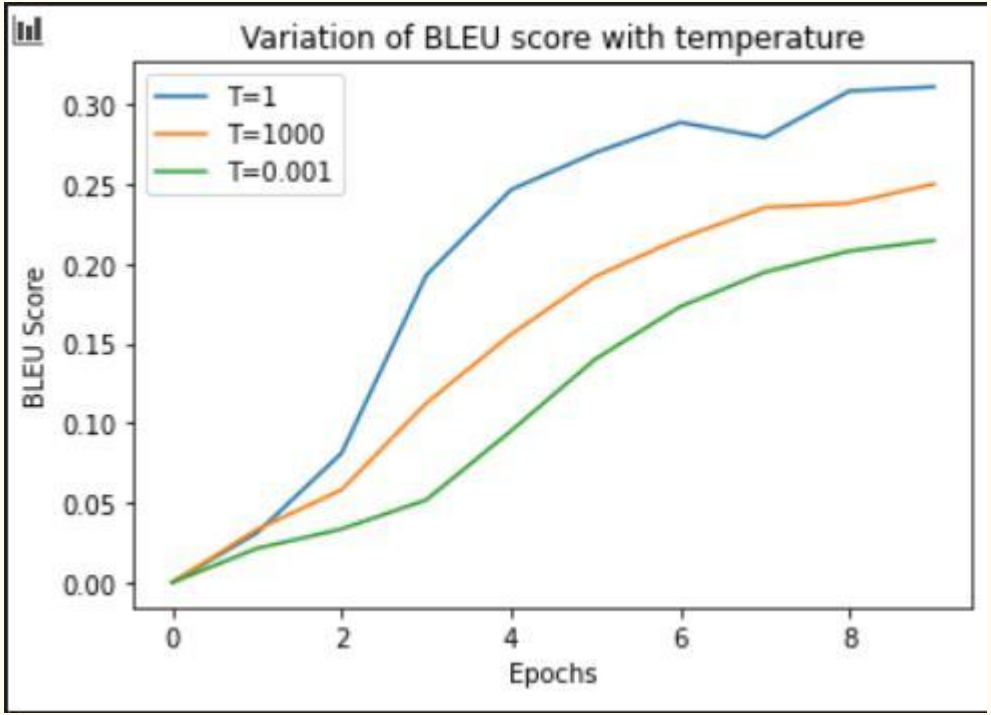
Experiment 1 - Effect of Attention and Adv. Training

- In the first setup, we analyse the effect of Adding attention and adding attention and adversarial training on our base model.
- We clearly observe the baseline model with no attention and no adversarial training performs poorly and only reaches Bleu score of about 25.
- We further observe, our adversarial training setup helped and has produced a little increase in the Bleu score as compared to the model with only attention.



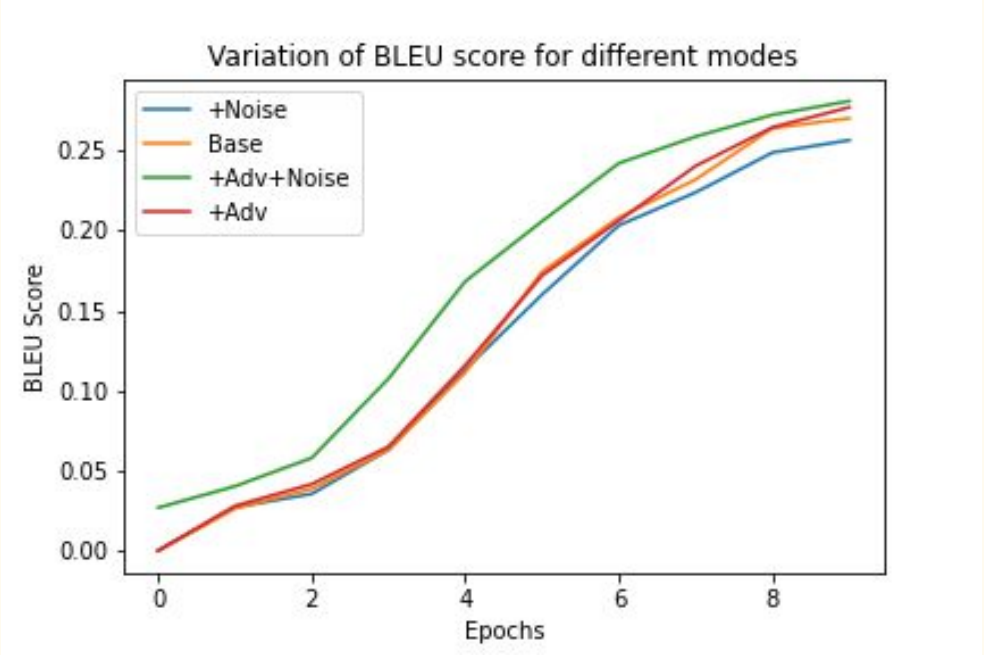
Experiment 2: Varying temperature to regulate amount of Attention

- In this experiment, we change the softmax temperature , which would in turn control the amount of attention and weightage we would be giving to each word while translating
- We see that a very high temperature, which means no attention or effectively, equal weightage for all words, achieves a Bleu score of just 25.
- At the same time, a very low temperature, which means very high attention to most probable word, also has a very low Bleu score of about 20.



Experiment 3: Effect of Adv. Training and Random Noise

- In this we analysed the effect of Noise and Adversarial training on our base model.
- There are 4 variations:1)Base,2) Adversarial Training, 3)Noise only, 4) Adversarial Training + Noise.
- Noise was added to encoder side hidden state, whereas the noise due to adversarial training was implemented at decoder side
- We can clearly observe the trend that, Adversarial + Noise happens to be best followed by only Adversarial, then Base and finally by only Noise. In short adding only Noise won't necessarily reduce the overfitting problem, we need adaptive noise too, like done in adversarial training.



Future Work

- To train our model and compare results on more bigger and standard language datasets.(like WMT14)
- The epsilon used in adversarial training can be made adaptive to embeddings. We have kept it fixed in our experiments.
- As the above described adversarial training works on RNNSearch it should also work with Transformers. We could apply the same method to improve the BLEU scores of SOTA Transformers.
- Handle unknown or OOV words in a better way. Specifically to handle those sentences where the frequency of words in corpus is less, as was seen in error analysis. Also solve the problem of multiple same words appearing in translations.
- Generate Attention maps and analyse their effect more coherently

Thank You