

### Assignment 3

CMSC 673 — Natural Language Processing

Faisal Rasheed Khan

VB02734

[vb02734@umbc.edu](mailto:vb02734@umbc.edu)

#### Question 1-----

1.

a.

$W, U, \theta$  are the weights for the RNN

$W$ - It is the weight matrix for the hidden state in the RNN cell. It helps in transform from previous hidden state to the current hidden state.

$U$ - It is the weight matrix which is used for the current input word and transforms to the current hidden state.

$\theta$ - It is the weight matrix for the output label word used in softmax function which transforms from current hidden state to current output label

b.

$h_i$  is a  $K$  dimensional vector, each embedding vector  $e_v$  is an  $E$  dimensional vector, and there are  $L$  possible label types

$W: K \times K$

$U: K \times E$

$\theta: L \times K$

c.

	1	2	3	4	5	6
$y_i$	The	can	can	hold	liquid	EOS
$w_i$	BOS	The	can	can	hold	liquid

d.

“The gray fluffy cat” be a sentence.

Teacher Forcing:

With teacher forcing,  $w_i$  input is always the actual input. What ever the probability is predicted whether it's the same predicted next word or different next word, the next input word is fed to the network whether or not the probability predicted word previously matches or not.

	1	2	3	4	5
Predicted word	The	<b>black</b>	fluffy	cat	EOS
Input word	BOS	The	<b>gray</b>	fluffy	cat

Not using Teacher Forcing:

With no Teacher Forcing,  $w_i$  input is whatever the model predicts the probability for the previous input. Here the model gets penalized more frequently .

	1	2	3	4	5
Predicted word	The	<b>black</b>	fluffy	cat	EOS
Input word	BOS	The	<b>black</b>	fluffy	cat

## Question 2-----

2.

a.

Consider the sentence “Running into the classroom, he was injured”

Here the word running without seeing the futher words from the current word Running, the parts of speech for Running is verb.

After considering the other words in the sentence, Running is adjective. Therefore the left-to-right processing is a potential disadvantage.

b.

This bi-directional approach concatenates both left-to-right and right-to-left representations. This approach can correct itself based on the future circumstances. The example in part a explains how it can be used for correcting the parts of speech by coming back from right-to-left because it considers both past and future contexts for correcting the predictions.

For the language modelling we predict the next word based on the previous words seen. We actually need to generate the words based on the probabilities and we don't actually require to modify the words from the future words.

c.

$x = \text{"cats eat"}$

$y = \text{"Noun Verb."}$

$e_{\text{cats}} = (0.75, 0.25)$

$e_{\text{eat}} = (0.3, 0.2)$

$$W = Z = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$U = Q = \begin{bmatrix} 0 & -1 \\ -1 & 0 \end{bmatrix}$$

$$\theta = \begin{bmatrix} \theta_{\text{Noun}} \\ \theta_{\text{Verb}} \end{bmatrix} = \begin{bmatrix} 0.7 & -0.4 & -0.3 & 1 \\ -0.2 & 0.35 & 0.4 & 1 \end{bmatrix}$$

$$l_0 = r_3 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

$f = \tanh$

$$l_1 = f(W l_0 + U e_{w1})$$

$$= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} * \begin{bmatrix} 1 \\ 1 \end{bmatrix} + \begin{bmatrix} 0 & -1 \\ -1 & 0 \end{bmatrix} * \begin{bmatrix} 0.75 \\ 0.25 \end{bmatrix}$$

$$= \begin{bmatrix} 1 \\ 1 \end{bmatrix} + \begin{bmatrix} -0.25 \\ -0.75 \end{bmatrix}$$

$$l_1 = \tanh\left(\begin{bmatrix} 0.75 \\ 0.25 \end{bmatrix}\right)$$

$$l_1 = \begin{bmatrix} 0.635 \\ 0.244 \end{bmatrix}$$

$$r_2 = f(Z r_3 + Q e_{w2})$$

$$= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} * \begin{bmatrix} 1 \\ 1 \end{bmatrix} + \begin{bmatrix} 0 & -1 \\ -1 & 0 \end{bmatrix} * \begin{bmatrix} 0.3 \\ 0.2 \end{bmatrix}$$

$$= \begin{bmatrix} 1 \\ 1 \end{bmatrix} + \begin{bmatrix} -0.2 \\ -0.3 \end{bmatrix}$$

$$r_2 = \tanh\left(\begin{bmatrix} 0.8 \\ 0.7 \end{bmatrix}\right)$$

$$r_2 = \begin{bmatrix} 0.664 \\ 0.604 \end{bmatrix}$$

$$l_2 = f(W l_1 + U e_{w2} )$$

$$= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} * \begin{bmatrix} 0.75 \\ 0.25 \end{bmatrix} + \begin{bmatrix} 0 & -1 \\ -1 & 0 \end{bmatrix} * \begin{bmatrix} 0.3 \\ 0.2 \end{bmatrix}$$

$$= \begin{bmatrix} 0.75 \\ 0.25 \end{bmatrix} + \begin{bmatrix} -0.2 \\ -0.3 \end{bmatrix}$$

$$l_2 = \tanh\left(\begin{bmatrix} 0.55 \\ -0.05 \end{bmatrix}\right)$$

$$l_2 = \begin{bmatrix} 0.5 \\ -0.049 \end{bmatrix}$$

$$r_1 = f(Z r_2 + Q e_{w1} )$$

$$= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} * \begin{bmatrix} 0.8 \\ 0.7 \end{bmatrix} + \begin{bmatrix} 0 & -1 \\ -1 & 0 \end{bmatrix} * \begin{bmatrix} 0.75 \\ 0.25 \end{bmatrix}$$

$$= \begin{bmatrix} 0.8 \\ 0.7 \end{bmatrix} + \begin{bmatrix} -0.25 \\ -0.75 \end{bmatrix}$$

$$r_1 = \tanh\left(\begin{bmatrix} 0.55 \\ -0.05 \end{bmatrix}\right)$$

$$r_1 = \begin{bmatrix} 0.5 \\ -0.049 \end{bmatrix}$$

$$h_1 = \text{concat}(l_1, r_1)$$

$$h_1 = \begin{bmatrix} 0.635 \\ 0.244 \\ 0.5 \\ -0.049 \end{bmatrix}$$

$$h_2 = \text{concat}(l_2, r_2)$$

$$h_2 = \begin{bmatrix} 0.5 \\ -0.049 \\ 0.664 \\ 0.604 \end{bmatrix}$$

$$p(y_1 | w_1, w_2, \dots, w_N) = \text{softmax}(\theta h_1)$$

$$= \text{softmax}\left(\begin{bmatrix} 0.7 & -0.4 & -0.3 & 1 \\ -0.2 & 0.35 & 0.4 & 1 \end{bmatrix} * \begin{bmatrix} 0.635 \\ 0.244 \\ 0.5 \\ -0.049 \end{bmatrix}\right)$$

$$= \text{softmax}\left(\begin{bmatrix} 0.1484 \\ 0.109 \end{bmatrix}\right)$$

$$\frac{e^{0.1484}}{e^{0.1484} + e^{0.109}} = 0.509$$

$$\frac{e^{0.109}}{e^{0.1484} + e^{0.109}} = 0.49$$

$$p(y_1 | w_1, w_2, \dots, w_N) = \begin{bmatrix} 0.509 \\ 0.49 \end{bmatrix}$$

$$\text{loss } y_1 = \log(0.509)$$

$$\text{loss } y_1 = -0.293$$

$$p(y_2 | w_1, w_2, \dots, w_N) = \text{softmax}(\theta h_2)$$

$$= \text{softmax}\left( \begin{bmatrix} 0.7 & -0.4 & -0.3 & 1 \\ -0.2 & 0.35 & 0.4 & 1 \end{bmatrix} * \begin{bmatrix} 0.5 \\ -0.049 \\ 0.664 \\ 0.604 \end{bmatrix} \right)$$

$$= \text{softmax}\left( \begin{bmatrix} 0.77 \\ 0.73 \end{bmatrix} \right)$$

$$\frac{e^{0.77}}{e^{0.77} + e^{0.73}} = 0.504$$

$$\frac{e^{0.73}}{e^{0.77} + e^{0.73}} = 0.495$$

$$p(y_2 | w_1, w_2, \dots, w_N) = \begin{bmatrix} 0.504 \\ 0.495 \end{bmatrix}$$

$$\text{loss } y_2 = \log(0.495)$$

$$\text{loss } y_2 = -0.305$$

### Question 3-----

3.

As the large language models (LLM) have been famous for performing very well in Natural Language processing tasks, The paper [Bender et al. \(2021\)](#) projects the large language models aspects and their impacts on having large data, if not used carefully could have Environmental and financial costs, potential risks and harm. The above impacts are because Large Language Models does not have any Natural Language Understanding and it only works with the linguistic form details. Large Language models have the benefits of having good architecture and are trained on large data which results in good performances. They perform good for text summarization, and Machine Translation tasks like converting English to another language, which is very challenging to do. If the data is very large, then the risk of documentation debt is there, and this need to be handled these risks by curating the data. Training large data requires more computing efficiency and consumes energy and emits co2 emissions. To improve the performance of the LLM further, we need more computing power which requires cost in addition to the carbon emissions. To handle the issue, computationally efficient hardware and algorithms are prioritized, and deploy the models with lower energy costs. The authors of the paper [Bender et al. \(2021\)](#) have also mentioned how the large training data might encode the stereotypes and derogatory associations that harm people. The large size of the data doesn't necessarily mean diversity because there are different views of the people which can be biased. Mostly the data will not be positive, it will have negative views. The dominant view is presented in the training model for LLMs

if the neutral view is not present in the data. LLM's can mispresent this information if its neutral perspective is not there. To handle this, we need to add bias to the large models and these are automated systems to remove toxicity towards a particular group, need to have prior understanding and engage in harmful systems to know about that so that LLM can't harm. Large data leads to documentation debt which should be reduced according to the budget. The risk LLM contains is the coherence of information generated from the data which can be meaningless. The risks are controlled by humans such as biased/dominated views, and derogatory words but LLMs can't control and output abusive language. One of the risks LLMs pose is large coherent texts with no ground truth. Another one is due to the Machine translation, representing everything wrong by translating one language to another. Risks are also because of the large number of parameters present in the model.

Based on all the above challenges faced, the authors of the paper [Bender et al. \(2021\)](#) suggest research with time, effort, and planning. Suggested that the environmental and financial costs of the model and biases should be carefully considered to output neutrally. Proper and optimal documentation within the budget which presents the details of the data, and stakeholders. They suggest realigning the research goals which affect the model training/improvements, because to improve a model it will cost more.

#### Question 4-----

4.

a.

The T5 model discussed is Multiple Inputs, Multiple Outputs ("sequence-to- sequence": with time delay). It is because of the encoder-decoder model used in the sequence-to-sequence, the input is sequence of the words, and the outputs are generated based on the relations and dependencies seen in the sequence and then the outputs are generated based on those considerations, where T5 encoder-decoder model also considers similar to that.

b.

(T5ForConditionalGeneration)T5 model with a language modelling head on top means that T5 is a model which generates after analyzing the entire input which is a Sequence-to-Sequence with time delay, and a language model is a conditional based generation tasks which predicts the next word based on the previous words and then so on. In pytorch we calculate it by taking the output from T5 and then sending the output to the linear layer with the help of softmax function in finding the probabilities

c.

"The cat is on the chair" into the French sentence "Le chat est sur la chaise."

The required arguments are input\_ids and labels.

The original English sentence input\_ids tokens are generated and the output French sentence labels tokens are generated to pass these variables to the forward function.

We apply a prompt for the input\_ids to indicate the translation task.

In the input\_ids tokenizer function we need to give information “translate English to French: “ by appending it to the input sentence and generate input\_ids for the appended sentence. For the labels we tokenize the sentence as it is given.

```
from transformers import T5Tokenizer, T5ForConditionalGeneration

tokenizer = T5Tokenizer.from_pretrained("t5-small")

model = T5ForConditionalGeneration.from_pretrained("t5-small")

input_ids = tokenizer("translate English to French: The cat is on the chair",
return_tensors="pt").input_ids

labels = tokenizer("Le chat est sur la chaise.", return_tensors="pt").input_ids

# the forward function automatically creates the correct decoder_input_ids

loss = model(input_ids=input_ids, labels=labels).loss

loss.item()
```

#### Question 5-----

5.

NLPAssign3co.ipynb

Used T5ForConditionalGeneration model for the GLUE RTE dataset. Prepend the sentence1 and sentence2 together to generate entailed or not. Fine-tuned the T5ForConditionalGeneration model. Used t5-small, evaluated model using the metrics accuracy, recall, precision and F1.

Ran the tasks using GPU in Colab and although it were university credentials, the usage of GPU got exhausted and I used limited number of sentences for the model because of the limitation of system resources.

Just ran with the very basic settings and also for less epochs that's why the accuracy is not that good.

```
accuracy 0.4
precision 0.16
recall 0.4
f1 0.2285714285714286
```

Just with the T5ForConditionalGeneration pretrained the accuracy is good compared to the former mentioned above.

```
accuracy 0.51
precision 0.506375
recall 0.51
f1 0.4612656267286204
```

## References:

[08-rnn-sequence.pdf \(umbc.edu\)](#)

[09-general-lm.pdf \(umbc.edu\)](#)


[T5 \(huggingface.co\)](#)

[Glossary \(huggingface.co\)](#)

[\[1910.10683\] Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer \(arxiv.org\)](#)

[Welcome to PyTorch Tutorials — PyTorch Tutorials 2.0.1+cu117 documentation](#)

[3.3. Metrics and scoring: quantifying the quality of predictions — scikit-learn 1.3.2 documentation](#)

Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. On the Dangers of Stochastic Parrots: Can Language Models Be Too Big? . In Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency (FAccT '21). Association for Computing Machinery, New York, NY, USA, 610–623.  
<https://doi.org/10.1145/3442188.3445922>