Name: Sachin Verma

Contact Number: 8930083406

Branch: Mechanical Engineering

Email Id: [sachin\_v@me.iitr.ac.in](mailto:sachin_v@me.iitr.ac.in)

Enrollment number: 22117122

**Project Title**: Reproducibility of NLP tasks with Human Gaze-Guided Neural Attention

**Selected Reasearch Paper:**

**Title:** Improving Natural Language Processing Tasks with Human Gaze-Guided Neural Attention

**Conference:** NeurIPS

**Link to Paper:** [**https://arxiv.org/pdf/2010.07891v2.pdf**](https://arxiv.org/pdf/2010.07891v2.pdf)

**Link to Paper Implementation:** [public-projects/human-gaze-guided-neural-attention-for-nlp: This repository contains the code for the work presented in the Neurips 2020 paper from Sood et al., 2020 - human-gaze-guided-neural-attention-for-nlp - Perceptual User Interfaces (uni-stuttgart.de)](https://git.hcics.simtech.uni-stuttgart.de/public-projects/human-gaze-guided-neural-attention-for-nlp)

**Reproducibility Summary**

**Scope of Reproducibility**: In this project, I have worked on the reproducibility of paper and analyzed its scope. This paper is for changing conventional way of reading comprehension tasks. It provides text saliency model for generating fixation data that can’t be obtained easily by conventional methods. In paper, it is been studied for sentence compression and paraphrase generation. I obtained the results for joint modelling of TSM with specific tasks and have done a ablation study also. But it can be studied for other comprehensions tasks with different generative architectures.

**Methodology:** I have used Pytorch 2 for the implementation of papers code to generate results. Other dependencies like sacrebleu for calculating bleu score, I have latest versions but they will not impact on the results of paper. I have used Kaggle GPU for training and validation and Google collab for predicting the output using required parameters.

**Results:** Paper studies two ways of training text saliency model for predicting fixation data. First one is to jointly train TSM for specific task model. Other one is to pretrain TSM to generate fixation data and use its weights for task specific model.  
I have studied first method and reproduced the results.

**What was easy:** Architecture of text saliency model along with the code was given along with used architectures for specific tasks. So it was easy to replicate the code for reproducing results.

**What was difficult:**

1. Code was given in a generalized context (except networks) without readme file. So it was difficult to understand the overall pipeline of model
2. Even the inputs and outputs of model were not clear. So majority of the time was spent for understanding and then writing code for the model.
3. I was using pretrained GloVe embedding file which was difficult to load on Kaggle everytime on starting kernel.
4. For pretraining TSM , a cognitive EZ reader model is used by authors. But data used by authors for training EZ reader model is not found. I have looked on data claimed by authors but it does not contain cloze predictability feature used by EZ reader.
5. I emailed authors for the dataset but was not replied back
6. It took around 80 hours for training(only) on Kaggle GPU so I was not in the condition to do unintuitive ablations.

**Introduction of the paper**:

This paper focuses on integrating human gaze data into neural network architectures for NLP tasks, specifically paraphrase generation and sentence compression. The researchers propose Hybrid text saliency model and two approaches to use it. The TSM predicts fixation durations during reading , then its predictions are integrated into attention layers of tasks-specific networks, each with different scores.

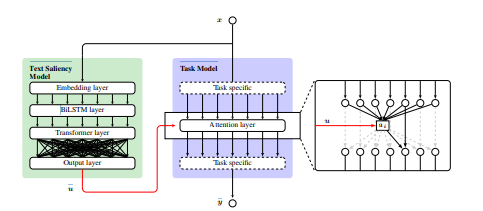
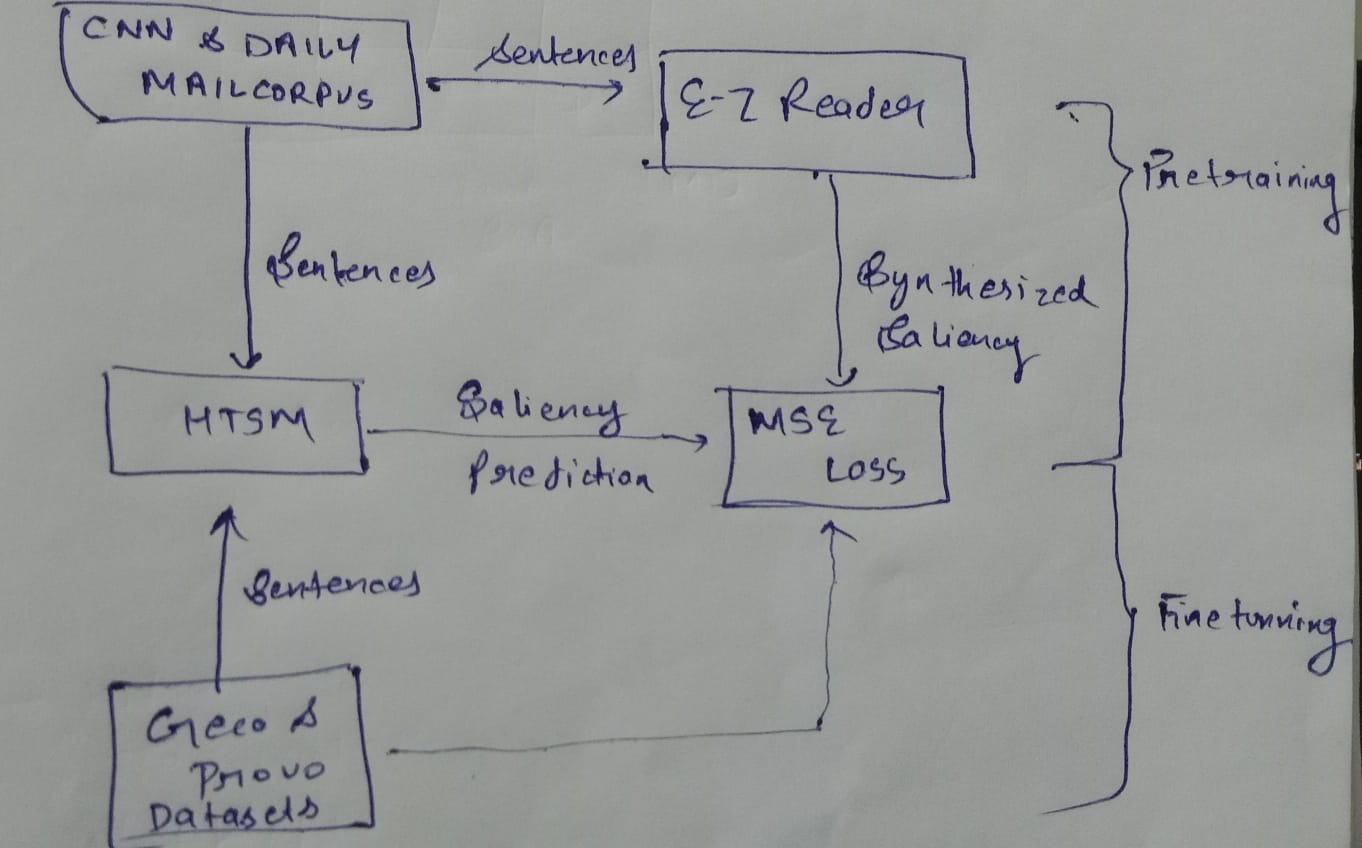


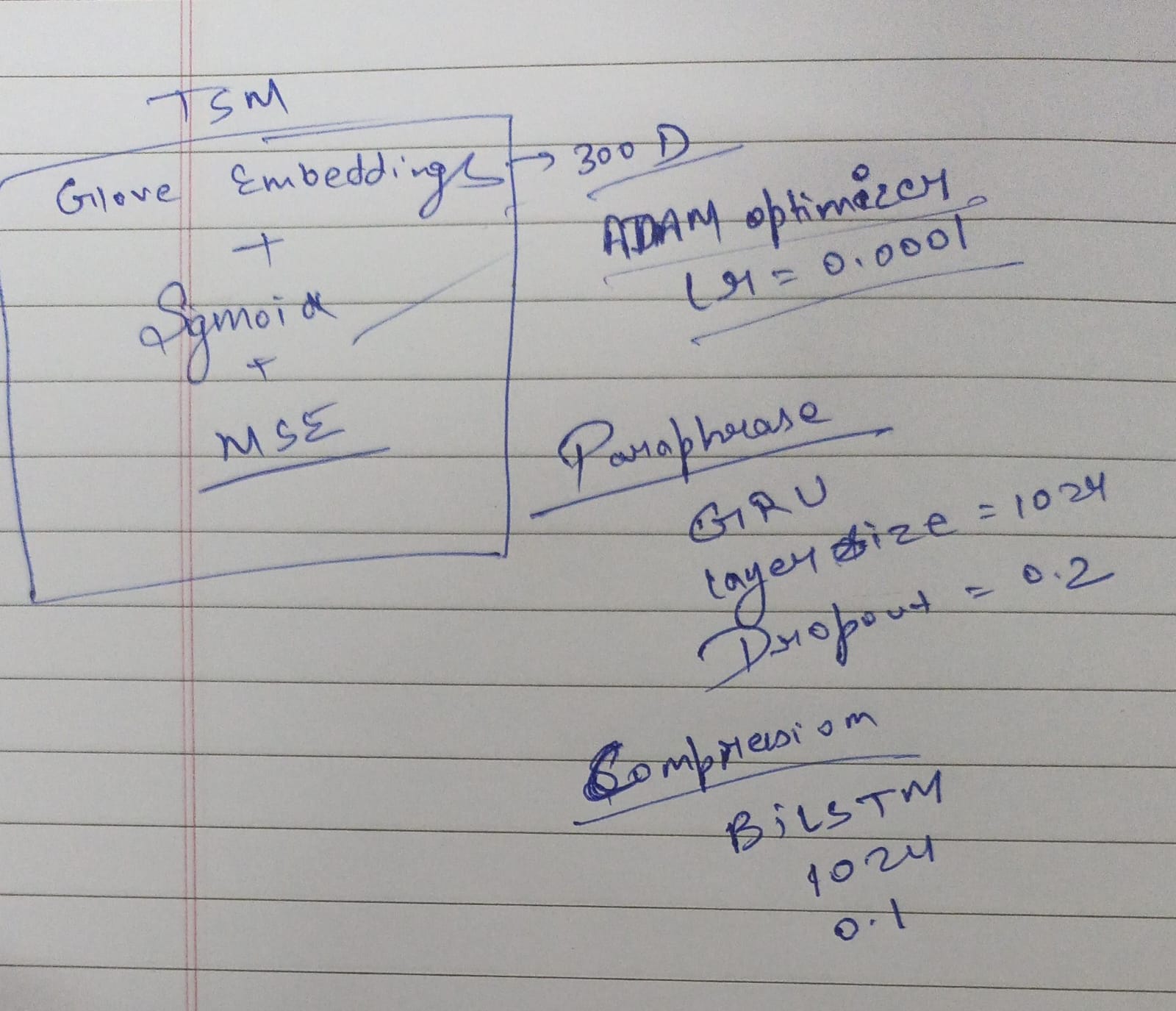
Figure 1 TSM and Joint model

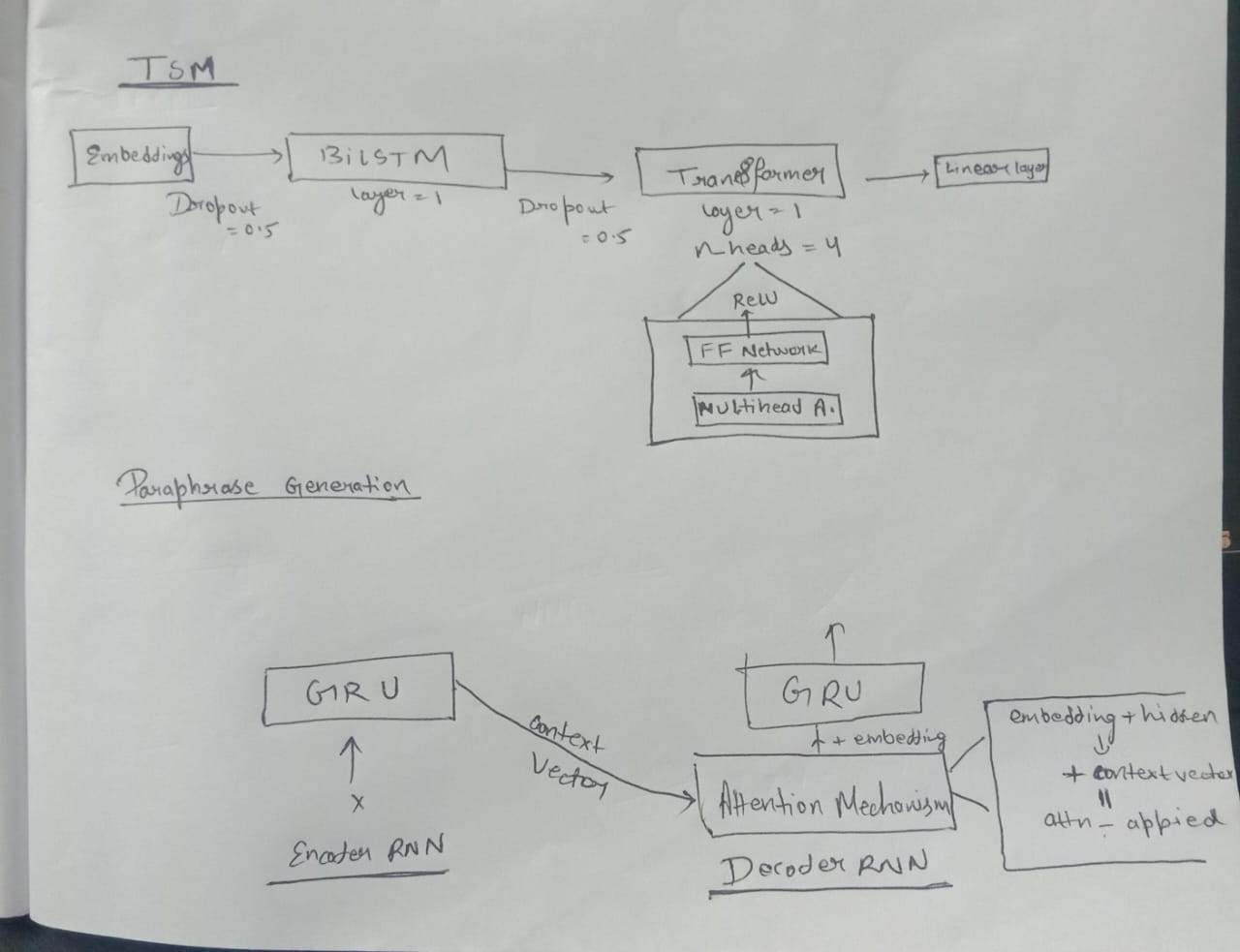
This TSM predicts fixation data using two approaches:

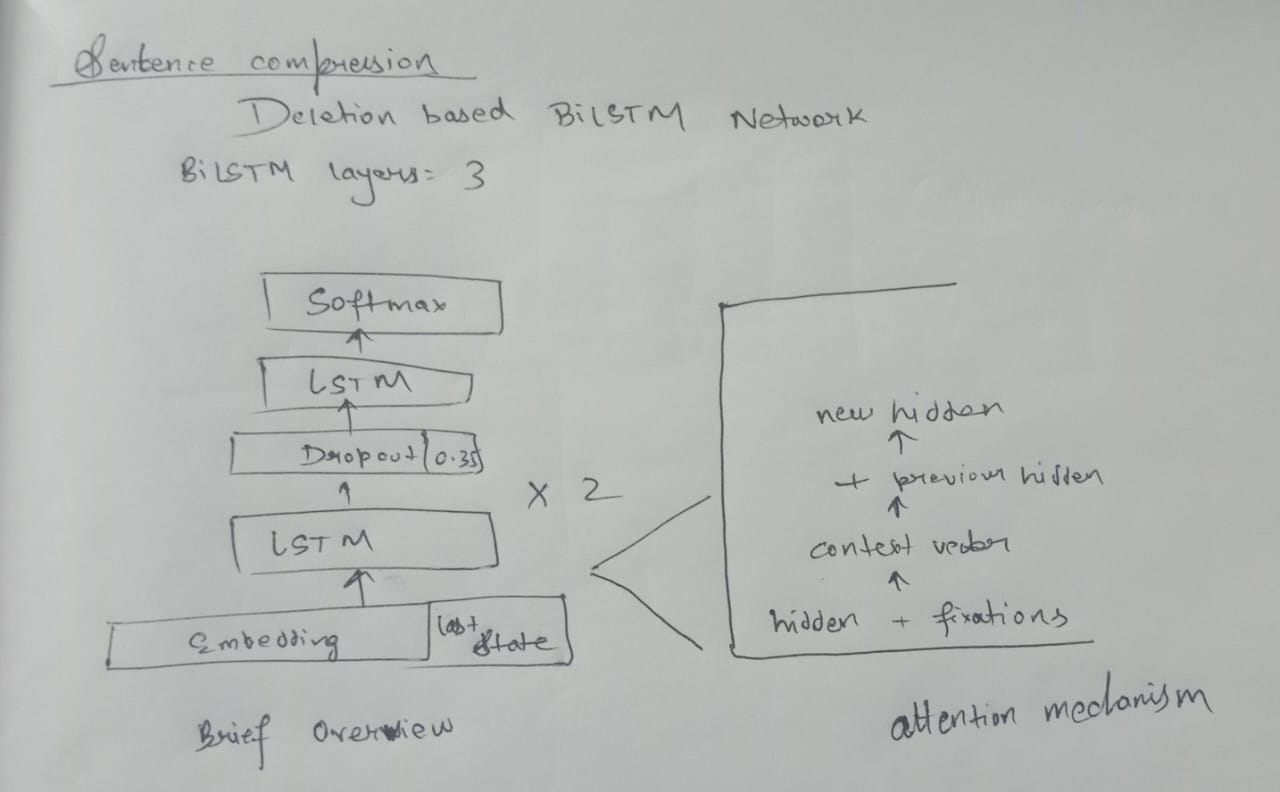
1. In first approach, researchers have used data provided by E-Z reader model (Cognitive approach model).



1. In second approach, they have jointly trained the TSM with a task-specific network, the saliency predictions are adapted to this upstream task without the need for explicit supervision using real gaze data







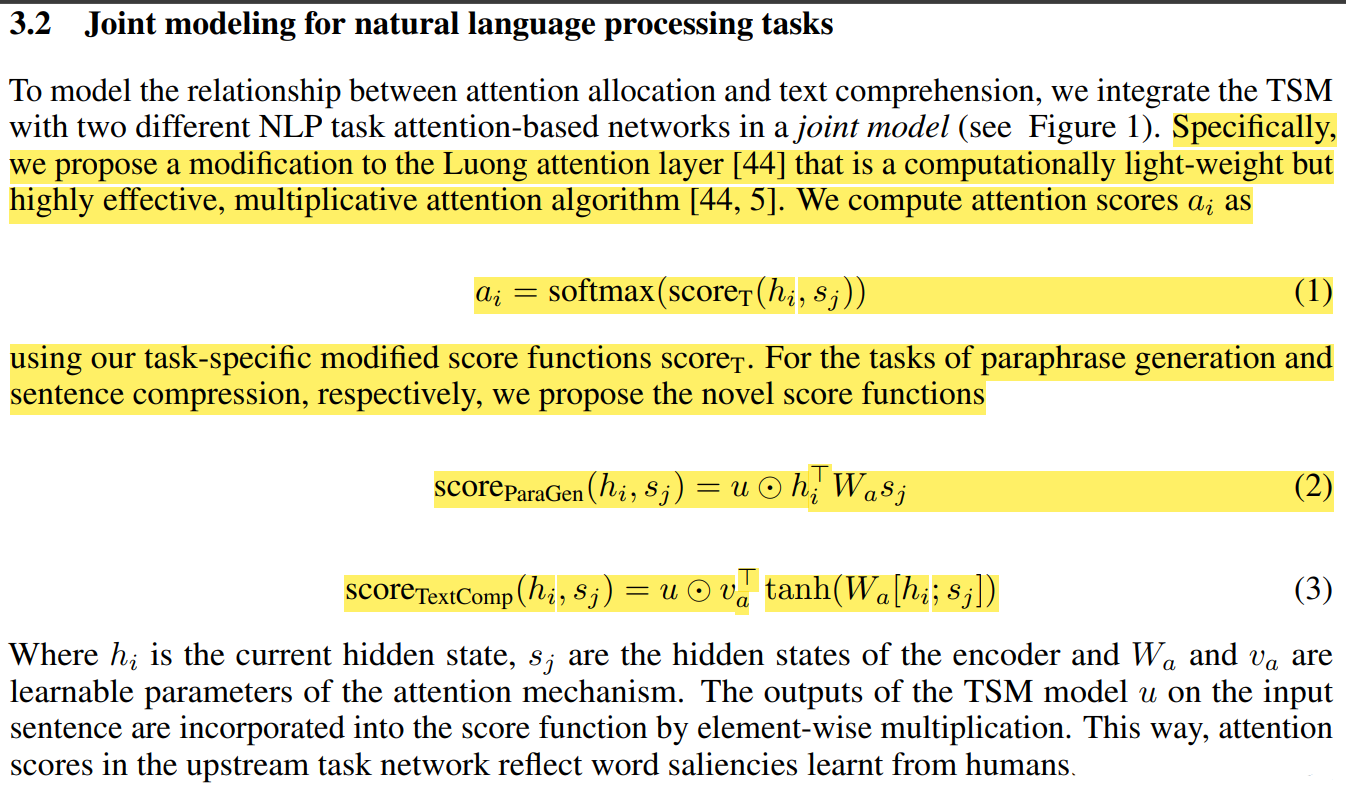


Figure 2 Score functions for each task

**Experimental Study:**

1. Hybrid Text Saliency Model (HTSM) Training

* **Data Sources**

Synthetic Data from E-Z Reader Model- To overcome the limitation of insufficient eye-tracking data for reading comprehension tasks.

Real Eye Tracking Data- In the second stage of training, the HTSM is fine-tuned using real eye- tracking data obtained from humans reading the Provo and Geco corpora.

1. Joint Modeling for Natural Language Processing (NLP) Tasks

* For paraphrase generation and sentence compression.
* Ablation studies include models without fixation data, freezing TSM, TSM weight swap

**Methodology:**

**Datasets:**

https://www.quora.com/q/quoradata/First-Quora-Dataset-Release-Question-Pairs https://github.com/google-research-datasets/sentence-compression https://cs.nyu.edu/~kcho/DMQA/ https://osf.io/sjefs/ https://expsy.ugent.be/downloads/geco https://www.perceptualui.org/publications/sood20\_conll/  
  
**Results:**

For paraphrase generation:  
Qqp Dataset: Training samples-50k

Validation samples-45k

Test samples-10k

Epoch 1



Epoch 12



TSM predictions

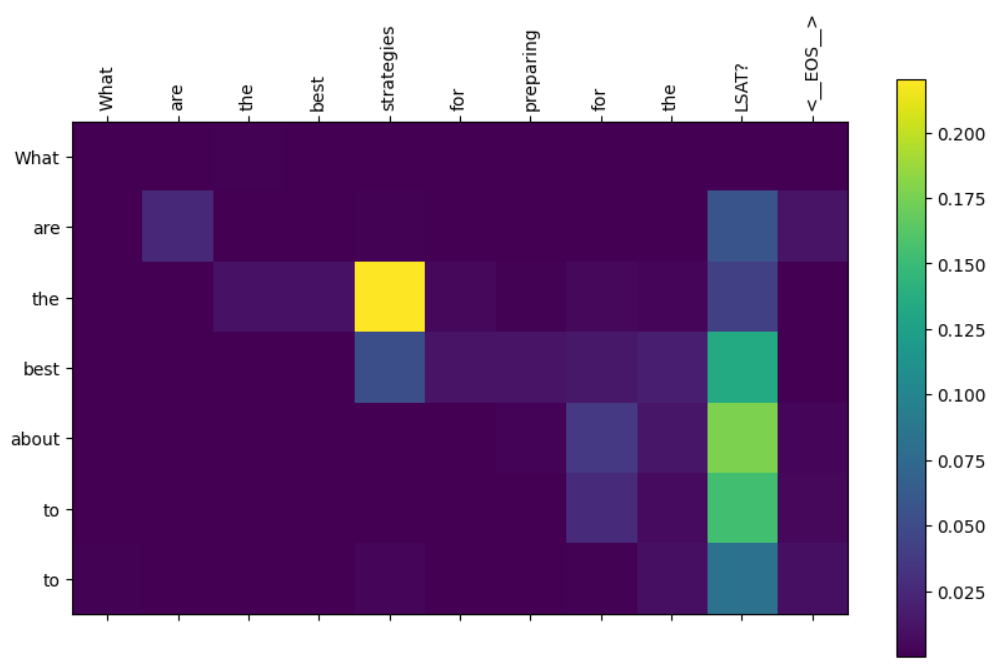


Figure 3Neural Attention Map

For Sentence Compression

Google sentence compression dataset: Train-89k

Validation-10k

Test-23k

Epoch 7



Epoch 3



TSM predictions

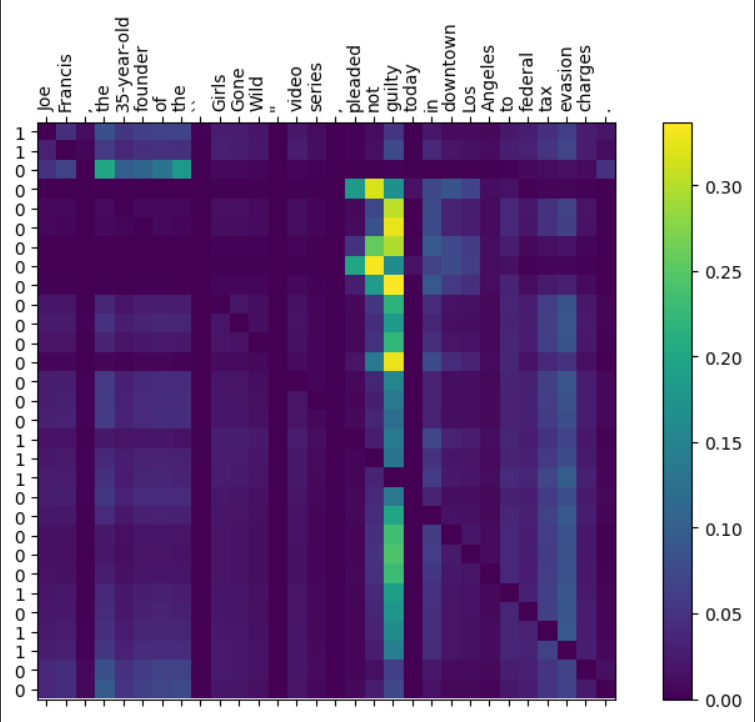


Figure 4Neural Attention Map

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Paraphrase Generation (BLEU-4) |  |  |  | Sentence Compression |  |
|  | 50k | Params |  | F1 | CR | Params |
| Score | 22.83 | 89M |  | 0.78 | 0.29 | 147M |

For 12 Epoch For 8 Epoch

**Ablations:**

1. I added positional encoding in my embeddings which is not added by authors and obtained scores for both tasks (BLEU for paraphrase generation and F1 for sentence compression)

I expected a increase in scores by adding positional encodings as they capture the information of relative position of words in the sentence.

But for paraphrase generation training is almost equivalent to baseline. However I noticed a significant increase in scores in sentence compression

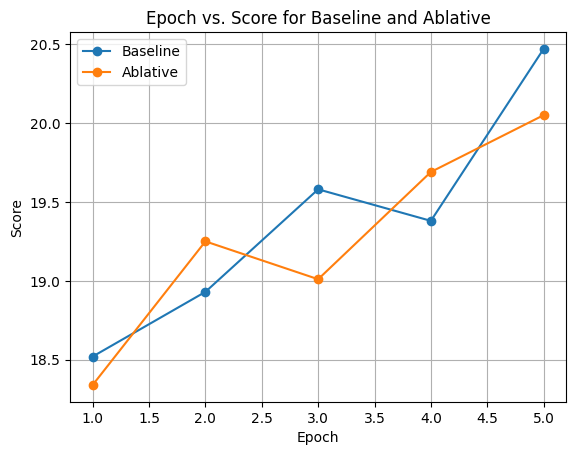
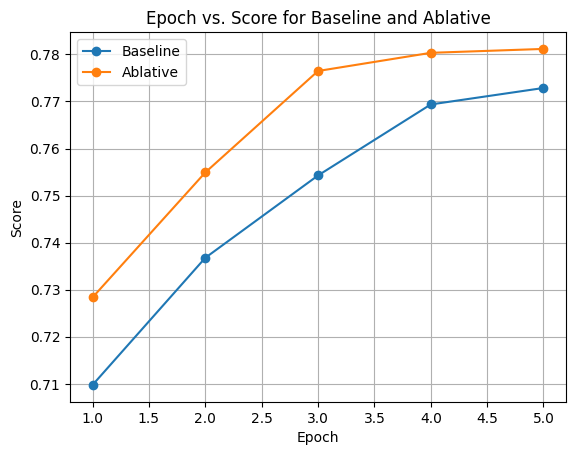
 

Figure 5Paraphrase Generation(BLEU) Sentence Compression(F1)

1. Authors have used GloVe embeddings which are non-contextualized. I implemented pre-trained BERT embeddings as they are contextualized but has been unable to train because it is taking too much time even on Kaggle GPU’s. I implemented on a portion of dataset and calculated a estimated time of 7 hours for tokenization and producing embeddings.