1 Question 1

The basic self-attention mechanism is a common approach to capture the semantic representation of words. It typically relies on creating a single attention vector, but this approach can be limited when applied to tasks such as sentiment classification, as it lacks sufficient input variety to capture diverse aspects of the sentence.

One approach to enhance the basic self-attention mechanism is to incorporate a max or average pooling layer over all time steps in the forward path. This can help aggregate features across time steps, but it might be challenging to implement within an RNN-based architecture, particularly when managing long sequences.

A more effective improvement can be achieved by modifying the input to the attention mechanism. For instance, instead of using a unidirectional approach, a bidirectional LSTM can be employed to produce a richer representation. This allows the attention mechanism to operate on a more informative representation, capturing both forward and backward dependencies in the sentence.

Furthermore, a known issue with basic self-attention is its tendency to focus on redundant information, where multiple attention heads may concentrate on the same parts of the input. To address this, Lin et al. (2017) [3] introduced a structured self-attentive mechanism that includes a penalization term. This term, based on the Frobenius norm, encourages the attention heads to focus on different parts of the sentence, promoting diversity and leading to more comprehensive sentence representations.

Incorporating these enhancements, such as bidirectional context and penalization for redundancy, can significantly improve the performance of the basic self-attention mechanism.

2 Question 2

According to the paper [2], The motivations for this change were leaded in addressing several limitations of recurrent models, particularly for tasks like machine translation and other sequence-based problems.

Recurrent models process sequences sequentially, meaning each time step depends on the previous one. This limits parallelism and can lead to inefficiencies, especially with long sequences. Self-attention allows the model to process all tokens in the input sequence simultaneously, leading to faster computations.

Furthermore, In contrast to RNNs, where each time step must wait for the computation of the previous one, self-attention enables parallel processing of tokens, significantly speeding up training and inference times. This is especially beneficial for tasks like machine translation, where large datasets are involved.

Additionnaly, Recurrent models often struggle with capturing long-range dependencies in sequences, as the information has to pass through multiple time steps, leading to vanishing or exploding gradient problems. Self-attention directly connects every token with every other token in the sequence, making it easier to capture dependencies over long distances.

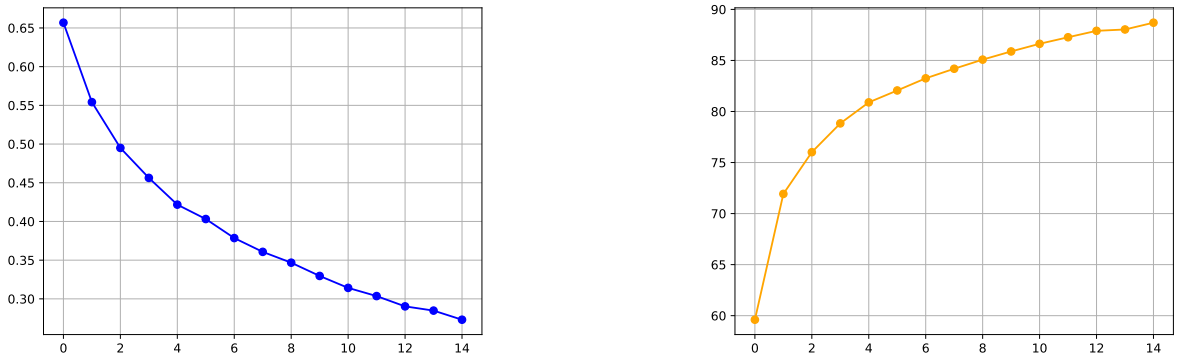
Transformers also reduced computational complexity, thanks to self-attention. Finally, RNNs and LSTMs can suffer from gradient vanishing or exploding when processing long sequences, making it difficult to learn long-range patterns. Self-attention mitigates this issue by enabling shorter and more direct paths for gradient flow during backpropagation, improving training stability.

3 Bonus (*purpose of my_patience parameter*)

In the context of training a neural network, the `my_patience` variable refers to the patience parameter used in early stopping during training. Early stopping is a technique used to prevent overfitting and stop training when a model's performance on the validation set stops improving. It can prevent also from overfitting.

4 Training process

I have trained the Hierarchical Attention Network during 15 `epochs`. The hyperparameters can be seen in the notebook.



(a) train losses over epochs.

(b) train accuracies over epochs.

Figure 1: train accuracies and losses during training, with 15 epochs.

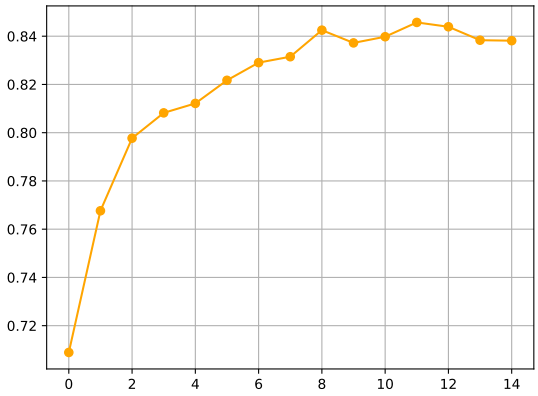


Figure 2: validation accuracies over epochs.

We can see figure 1 the train losses and accuracies during training, wich demonstrate a good training process. We can also observe figure 2 the validation accuracies over the epochs.

5 Question 3

For this question, I selected the last document from the test dataset, where the model predicts it to be a positive (”yes”) review. The sentence attention scores (Figure 3) show that certain sentences are given higher weight, likely because they express clear positive sentiment. For example, sentences like *”a masterpiece”* or either *”[...] downright brillant”* are prioritized by the model, as they play a key role in forming the positive classification.

The word attention scores (Figure 4) for the sentence *”:) First of all, Mulholland Drive is downright brillant.”* highlight words such as ”downright” or ”brillant,” which strongly contribute to the positive sentiment if they are putting together.

In conclusion, both sentence- and word-level attention scores indicate that the model identifies the most sentiment-relevant portions of the review, which explains its prediction of a positive (”yes”) classification.

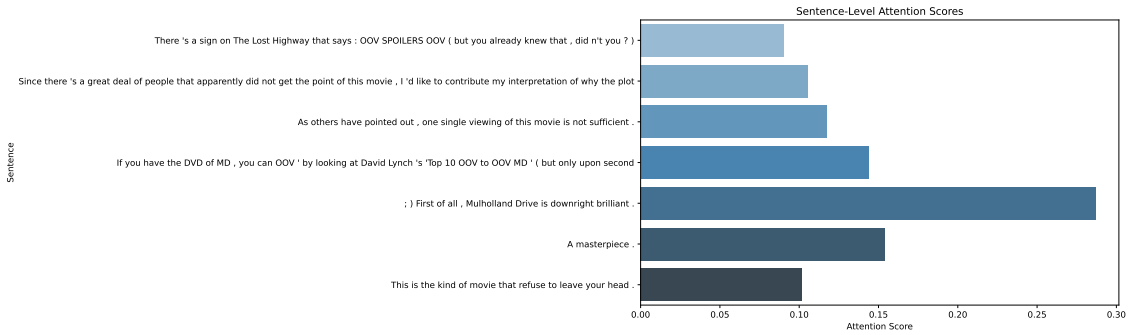


Figure 3: Sentence attention scores for the last document of the test dataset.

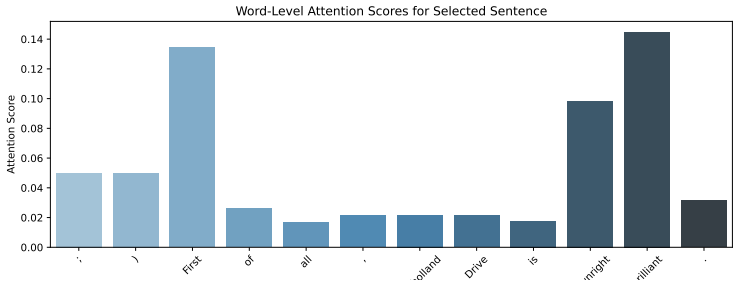


Figure 4: Word attention scores for the last sentence of the last document of the test dataset.

6 Question 4

According to the article [1], Hierarchical Attention Networks (HAN) have several limitations related to both architecture and training.
Firstly, the model processes sentences independently during the initial step of the architecture. This means that while one sentence is being analyzed, the model ignores other sentences, which can hinder its ability to capture the overall meaning of the document. As a result, the model may struggle with understanding context and relationships between sentences.
Additionally, the embedding representation for each sentence is uniform across all instances, which limits the extraction of diverse and complementary information. By not differentiating embeddings for various instances, HAN may miss important nuances that could enhance the model's performance on document understanding tasks.

References

[1] Jean-Baptiste Remy, Antoine Jean-Pierre Tixier, and Michalis Vazirgiannis. Bidirectional context-aware hierarchical attention network for document understanding, 2019.

[2] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023.

[3] Minwei Feng Bing Xiang Bowen Zhou Yoshua Bengio Zhouhan Lin, Cicero Nogueira dos Santos. A structured self-attentive sentence embedding. *ICLR*, ibm(1703.03130), 2017.