Celine Hajjar
celinehajjar98@gmail.com

1 Question 1

The basic self-attention mechanism described in the lab aims to extract an attentional vector for a given sentence by computing a weighted sum of the annotations of its different words. The main limitation of this approach is that the weights rely solely on one vector, the context vector, which can be interpreted as a representation of the optimal element in the sentence. However, one optimal element might not be enough to understand the semantics of long and complicated sentences. In [1], the authors bring up this issue and propose a method to solve it. It consists in replacing the context vector by a context matrix, where each row holds information about one part of the sentence. They also propose a way to avoid redundancy between the matrix rows by adding a penalization term so that attention weights focus on different aspects of the sentence.

2 Question 2

Up until recently, the fields of language modeling and machine translation were dominated by recurrent neural networks, and in particular by long short-term memory networks. One benefit of this type of networks is that unlike regular neural networks, they are able to capture the sequential nature of natural language. However, RNNs and their variants have been replaced by a more recent framework for language-related tasks, namely, the attention mechanism, for three main reasons, as presented in the original paper[3]:

1. Even though recurrent units capture the sequential relationship between different words in a sentence, they are sensitive to the length of the given sequence. Often, two words in a sentence can be related even if they are not close to one another, but this information risks being dissipated across time-steps when calculating hidden states in a recurrent network. Attention mechanisms solve this problem by processing all the annotations of the input sequence at once, not just the last one.
2. Due to the intrinsic sequential nature of RNNs, they are hard to train on a parallel system such as a GPU. However, the computation of the weights in the case of the attention mechanism can be done for all the inputs in parallel. This significantly decreases the computation complexity of the whole process.
3. Lastly, attention weights give the model some kind of interpretability: as we can see at the end of this lab, visualizing the attentional coefficients of a given review helps us understand which words were the most important for the classification of the document.

3 Question 3

I chose to plot the attention coefficients of the same test review used in the code. For each sentence, I made a bar plot representing the coefficients of the different words, sorted in a descending order (figure 1).
When it comes to sentiment analysis, the general feeling of the sentence can be inferred from a selection of words. This is confirmed by the plots. For example, in sentence 5, the word ”brilliant” has a significantly higher score than the other words in the sentence, and even in the whole review. The same goes for the word ”masterpiece” in sentence 6. However, the coefficients are less logical in some cases. In sentence 3, for example, the word ”is” is the most important one in the sentence, even though it is a very common word. Also, the polarizing words have a clear dominance over others, but when the sentence does not contain such words, the coefficients tend to vary less between words (sentence 3).

4 Question 4

In the first level of HANs, each sentence is encoded into a representation vector. However, this representation is computed in an isolated manner, meaning that other sentences are not taken into account. This ”lack of communication”, as stated in [2], is not optimal because the network cannot transfer knowledge between sentences. For example, if the same words are repeated in every sentence, the network will still compute attention weights for the same features every time, instead of focusing on other aspects.

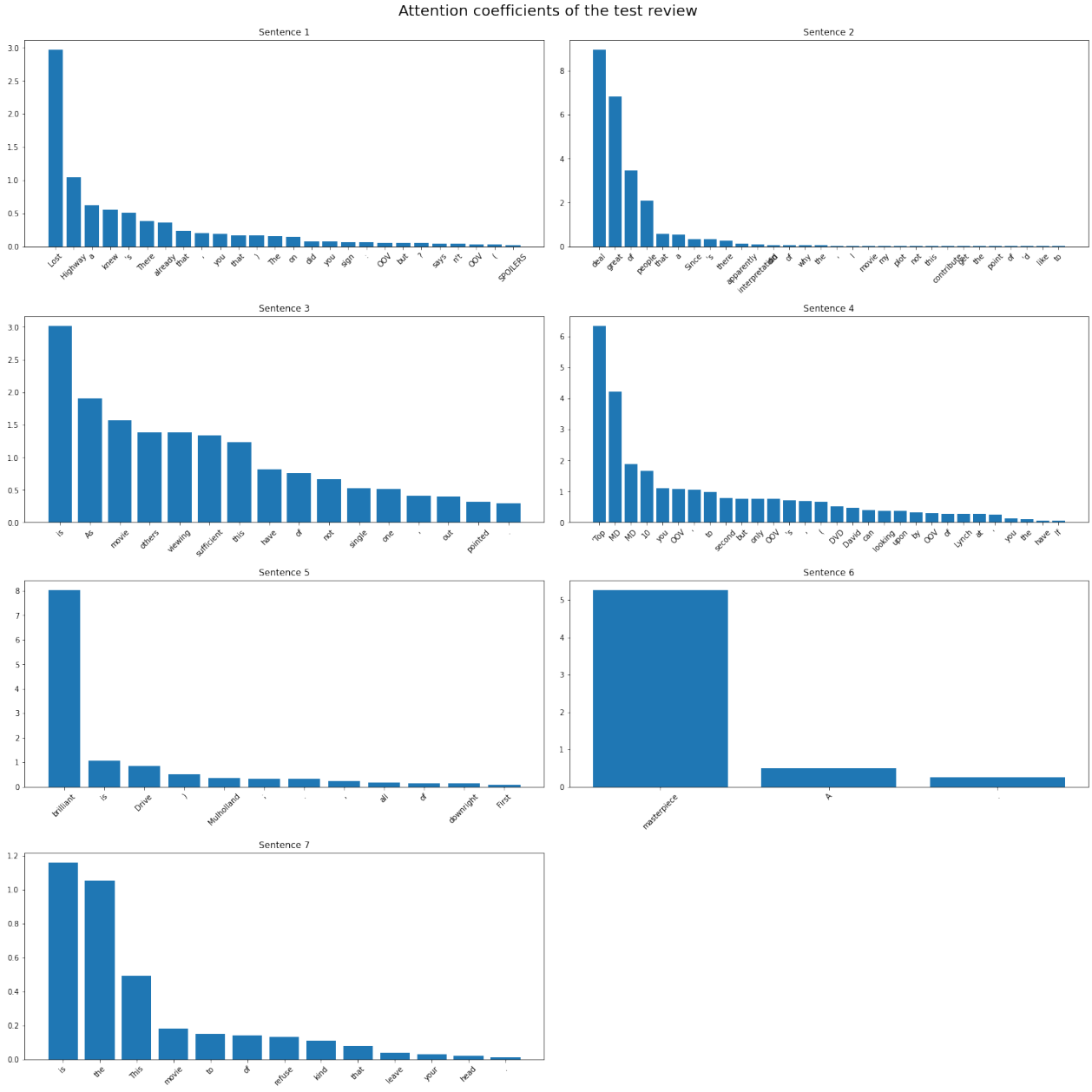

Figure 1: Attention coefficients of the words in the test review, by descending order

References

[1] Zhouhan Lin, Minwei Feng, Cícero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. A structured self-attentive sentence embedding. *CoRR*, abs/1703.03130, 2017.

[2] Jean-Baptiste Remy, Antoine Jean-Pierre Tixier, and Michalis Vazirgiannis. Bidirectional context-aware hierarchical attention network for document understanding. *CoRR*, abs/1908.06006, 2019.

[3] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.