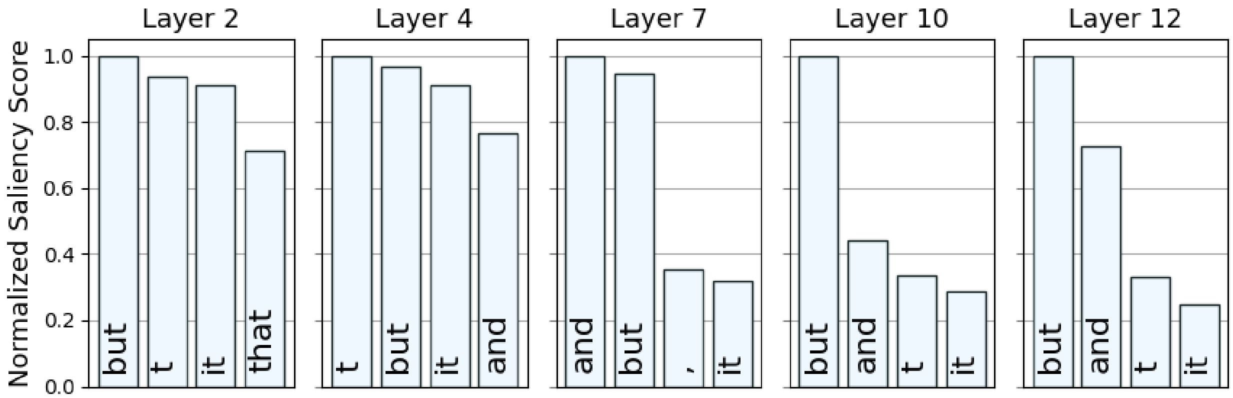
Results and methodology

First, we will look into the coordinate analysis task, this task's purpose is to analyze a multivariate dataset, to find a proximity matrix, getting a proximity matrix has major advantages for us, from domain reduction to prediction and more.

To gain these data we used a pre-trained BERT model to achieve probing features, these features help us to compute the importance of data (by saliency (how a human thinks about the importance of these data)), and this importance is a metric that can be used to calculate proximity matrix of data.

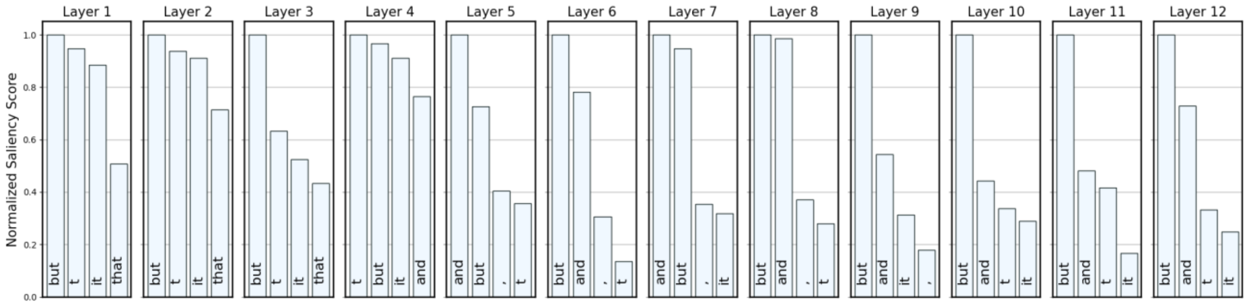


By running the codes the table above is generated, as it is clear the table represents the top 4 most important words in 5 different layers.

It is concluded that:

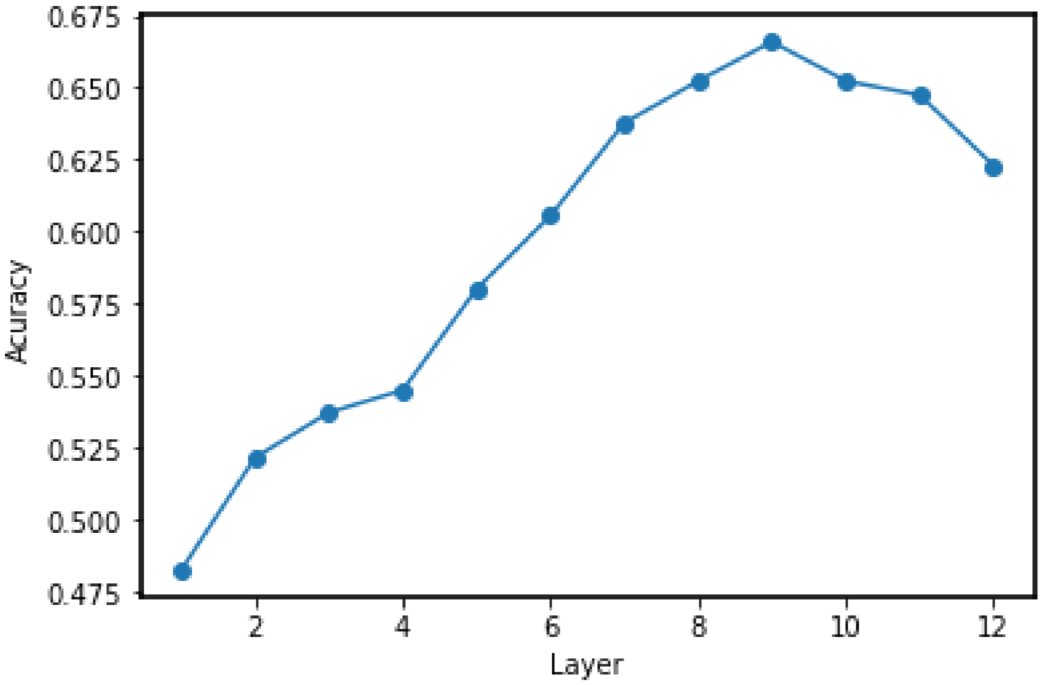
1. In layer 2, “but”, ”t”, ”it”, and ”that” are the most important.
2. In layer 4, “t”, ”but”, ”it”, and ”and” are the most important.
3. In layer 7, ”and”, “but”, ”’”, and ”it” are the most important.
4. In layer 10, “but”, ”and”, ”t”, and ”it” are the most important.
5. In layer 12, “but”, “and”, ”t”, and ” it” are the most important.

A more general graph is featured below



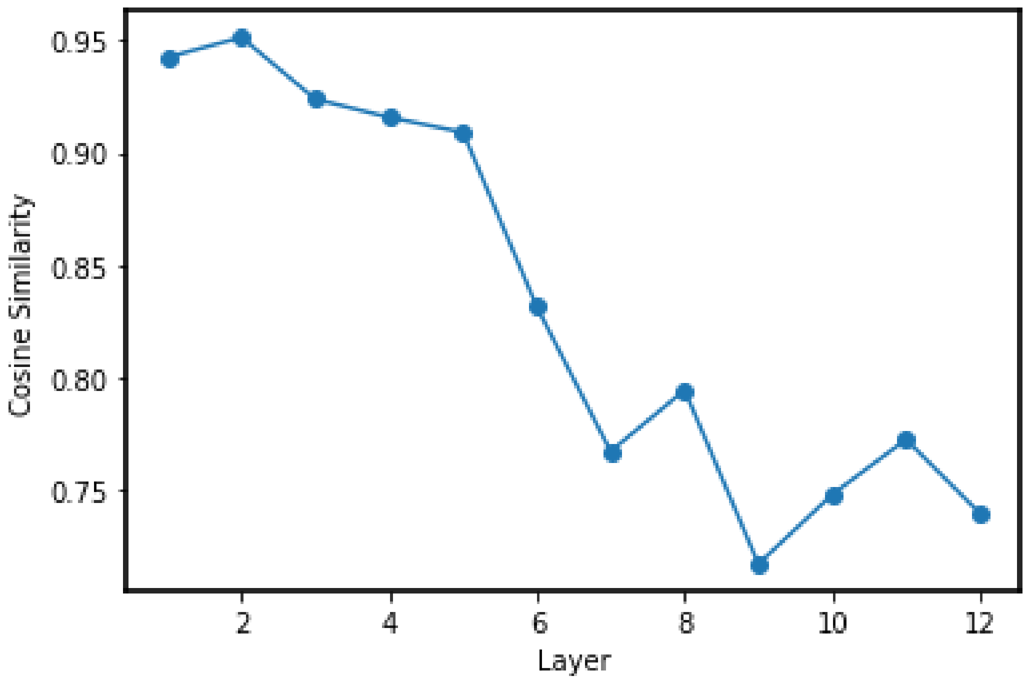
These words were the most important tokens by saliency that were featured in the representation of each layer that is mentioned in the graph.

But not all of the words that were considered important by the network, were really important! To measure the accuracy of pre-trained BERT in this task, the accuracy graph is plotted below:

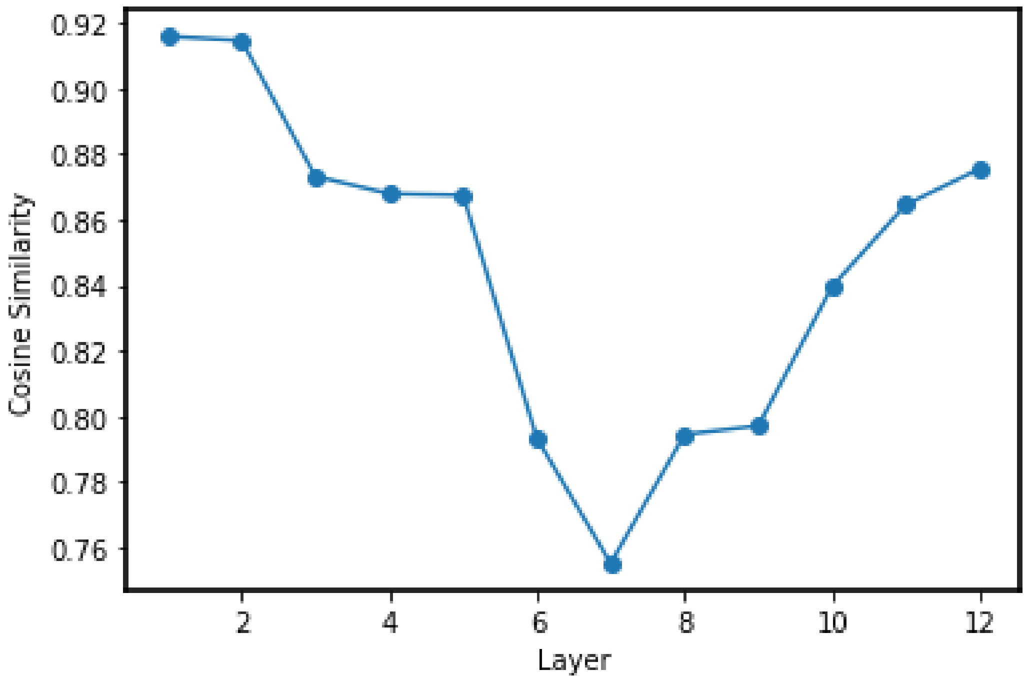


As it is clear, we can see that the middle layers (8 and 10) were the main layers that we can rely on to acquire the importance score of each word.

Another metric is to compare the similarity of results to BUT and AND (two of the ord that a human considers the most important), the graphs are below:

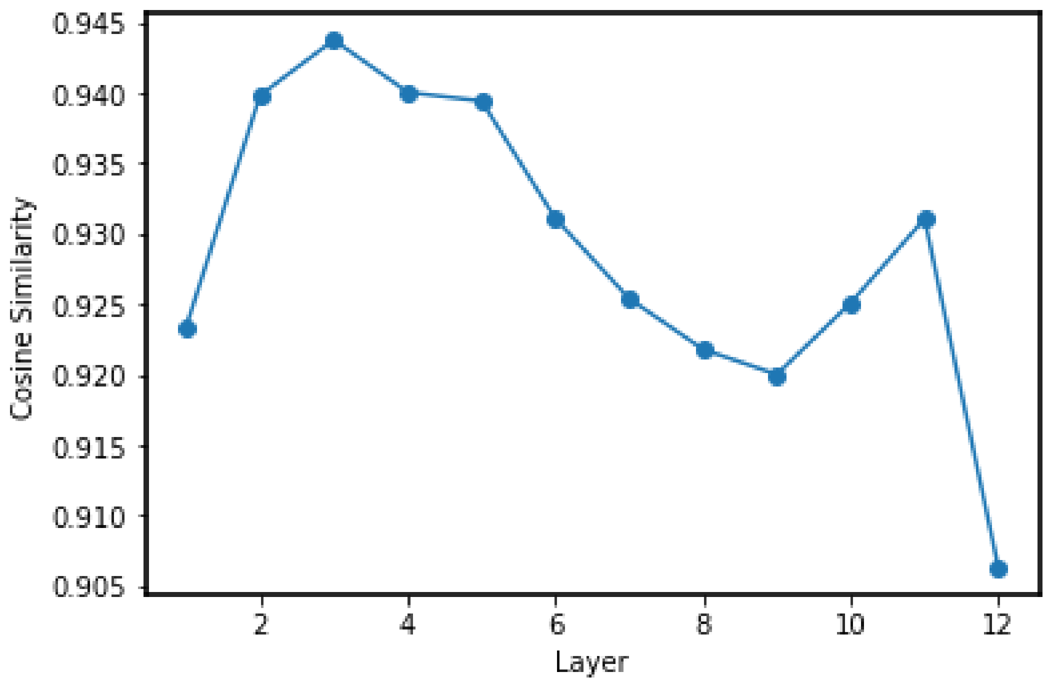


For BUT

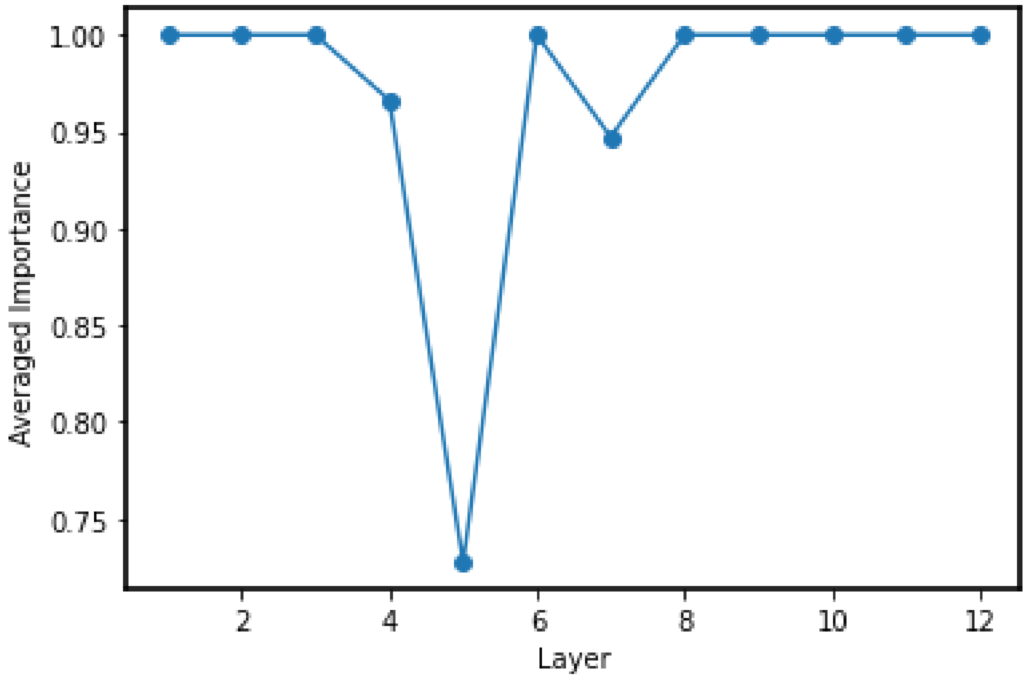


For and

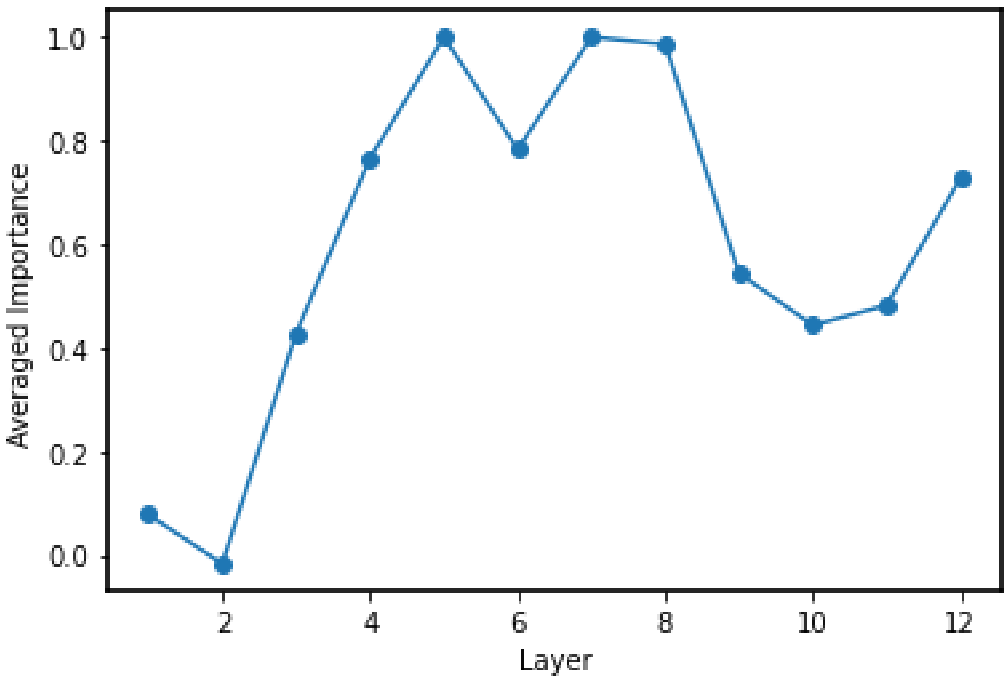
Also, another similarity graph is generated to compare word with a false (less important word like non-but (not but)). The graph is below:



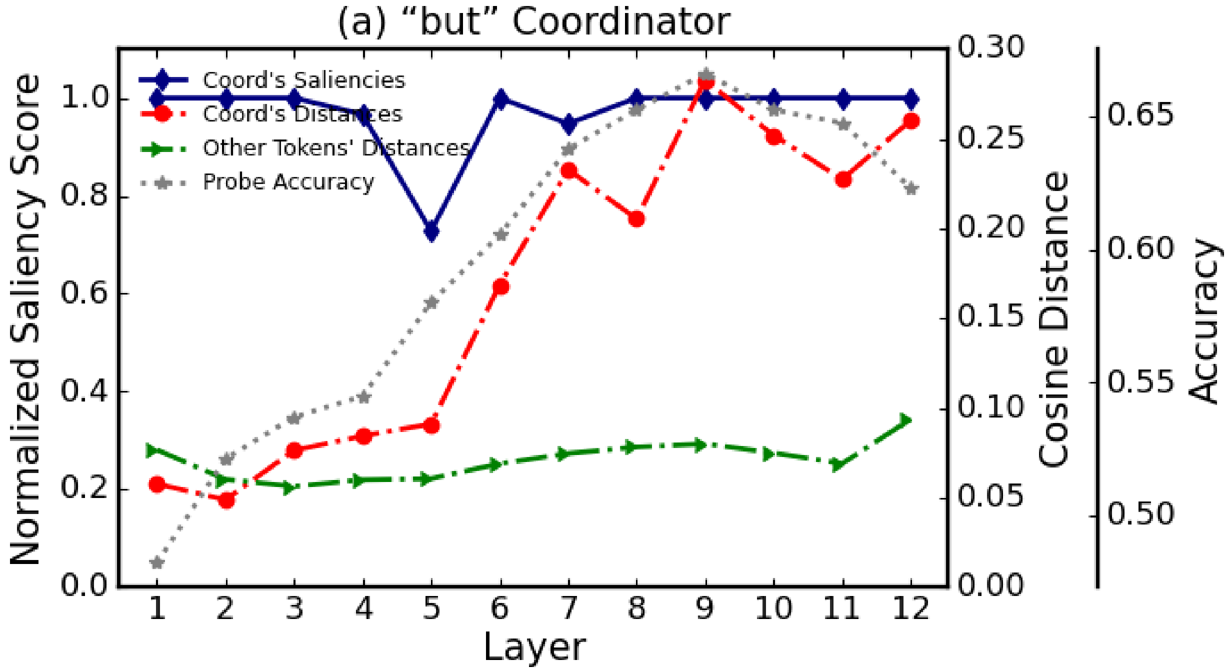
There were also normalized and integrated and – but graphs that are shown below:



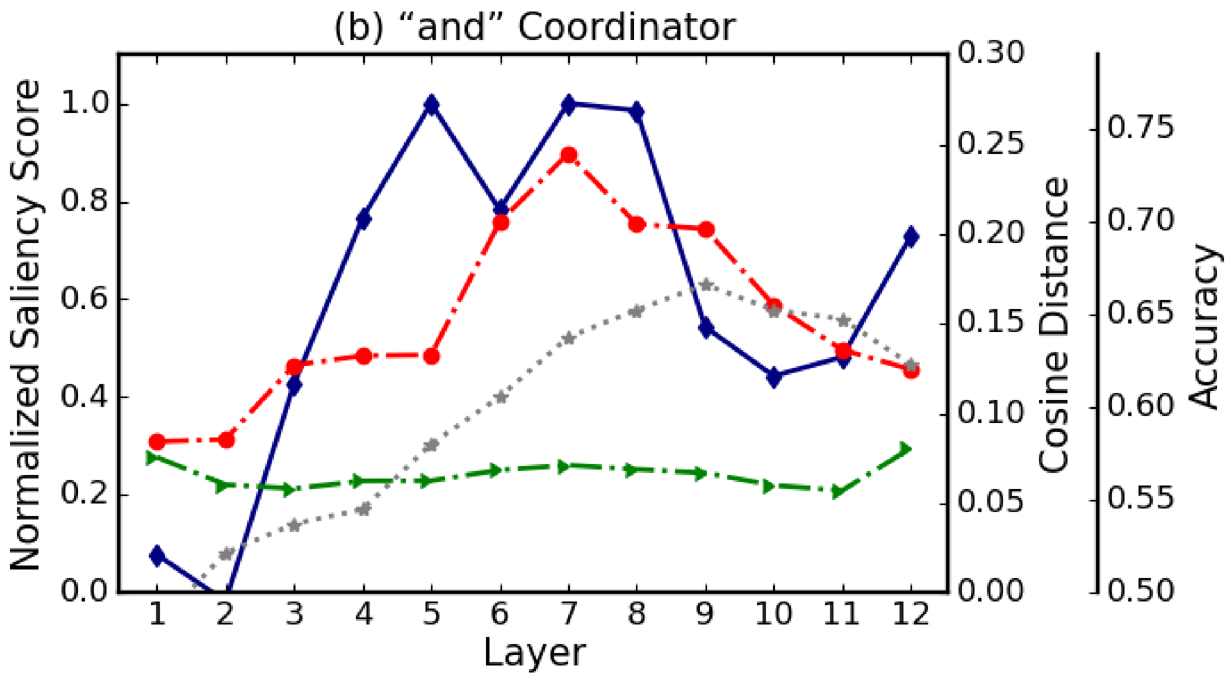
Normalized but



Normalized and



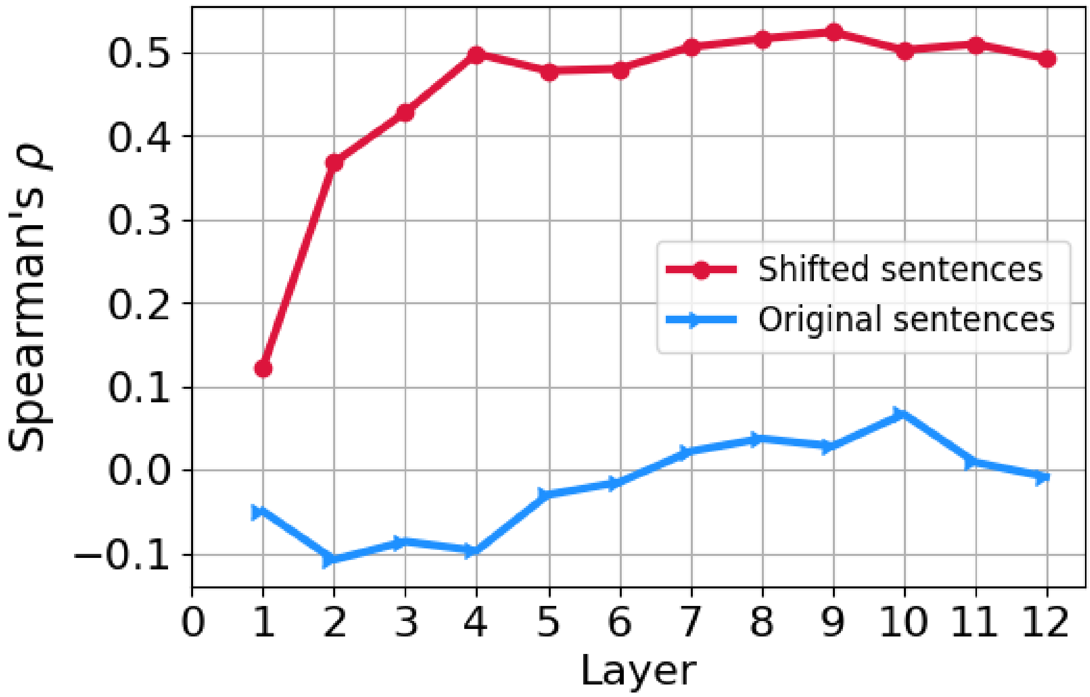
Integrated but

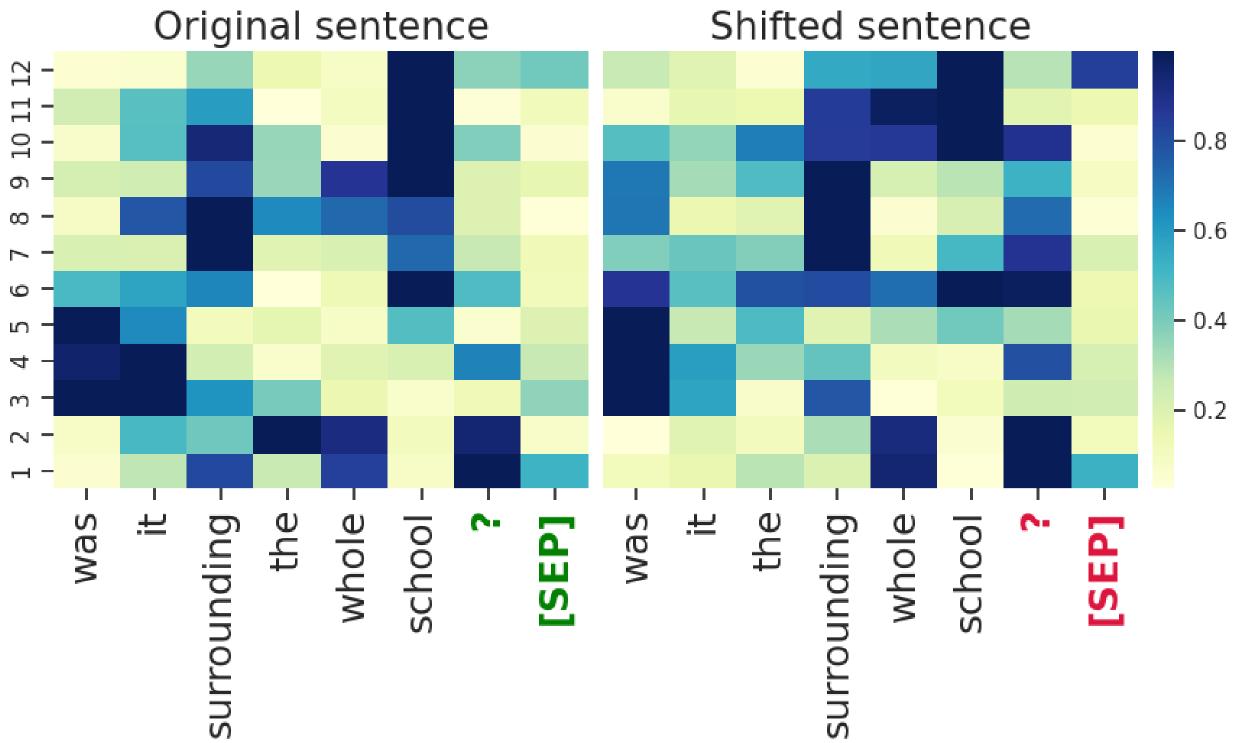


Integrated and

The next experience was BShift, this task was to measure error handling. In this task, all sentences were shifted and examined how Bert can handle this anomaly.

The same methods were followed in this experience, we acquired features, transferred learning to developed keras networks, and then used saliency evaluation.

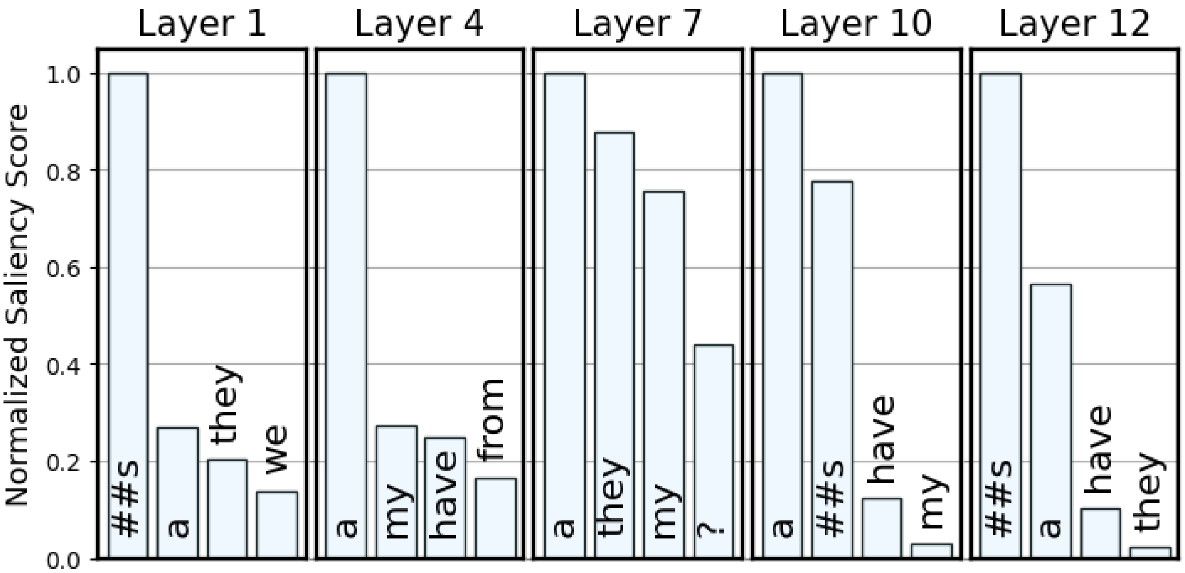




As it is clear, there is a large mse, between original and shifted sentences, and Bert could not handle this error very well.

OBJ num: in this task numbers are important, we want to examine which tokens occur most in a text or sentence. This task is a very important and known task that is used in probability and most of the NLP tasks!

As the method in most of the experiences are the same I am not going to explain that again (unless it is different).



The only graph that was represented in this task code was the above histograms.

As it is clear "s" (##s) and a are the most common words in English, which is True. As it is clear all of the layers mentioned in the histogram have achieved this result, but the last layers have better performances.

It is clear that:

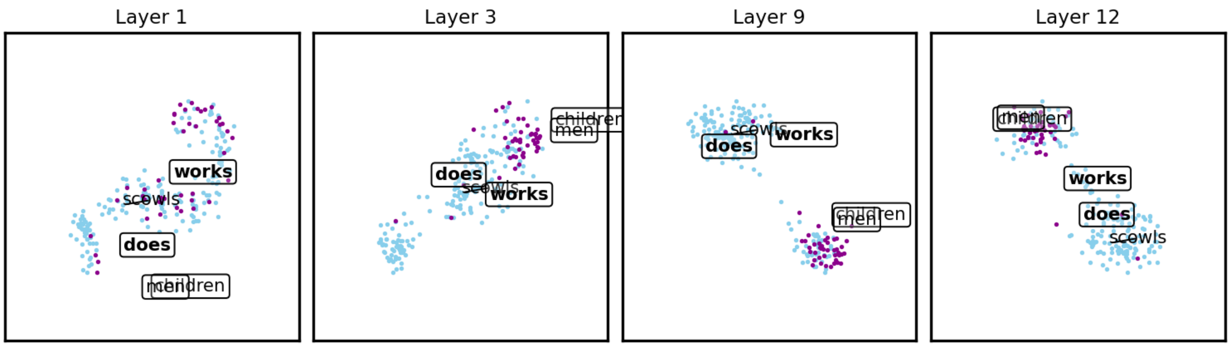
1. The first layers representation for objnum is "##s", "a", "they", "we"
2. The 4 layers of representation for objnum are "a", "my", "have", and "from". This layer has no "##s" in its representations.
3. The 7 layers representation for objnum is "a", "they", "my", and "?". This layer has no "##s" in its representations either.
4. The 10 layers representation for objnum is "a", "##s", "have", "my"

5. The 12 layers representation for objnum is “##S”, ”a”, ”have”, ”they”

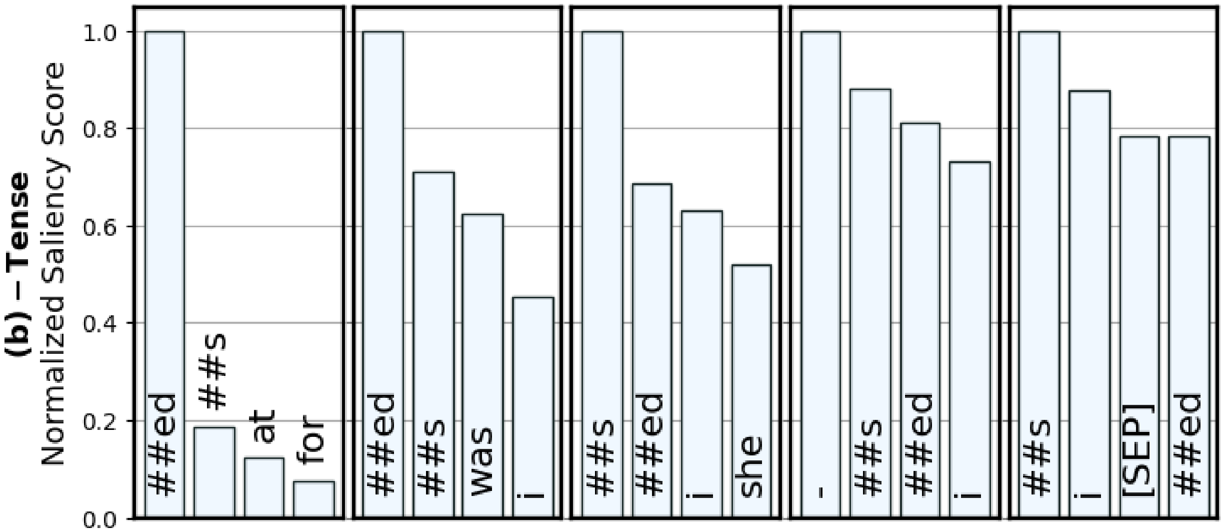
Tense: in this experience, as the name says everything, we are going to measure the tense and time of the words.

The difference in the methodology of this task is that it measures the TSNE for the data. But not for domain reduction, but for better representation.

The results of TSNE are below:



And the results of the layers are categorized in below:



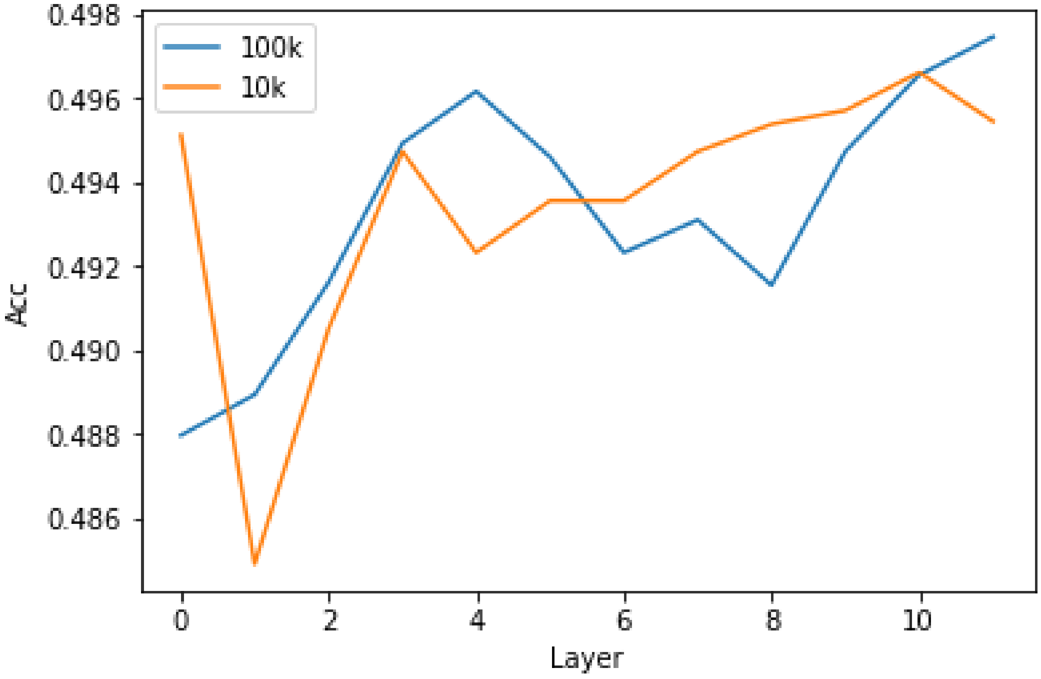
Bert has good performance for this task too, as it is clear I first glance , the most featured tokens are “##ed” and “##s” which is true.

We can clarify that:

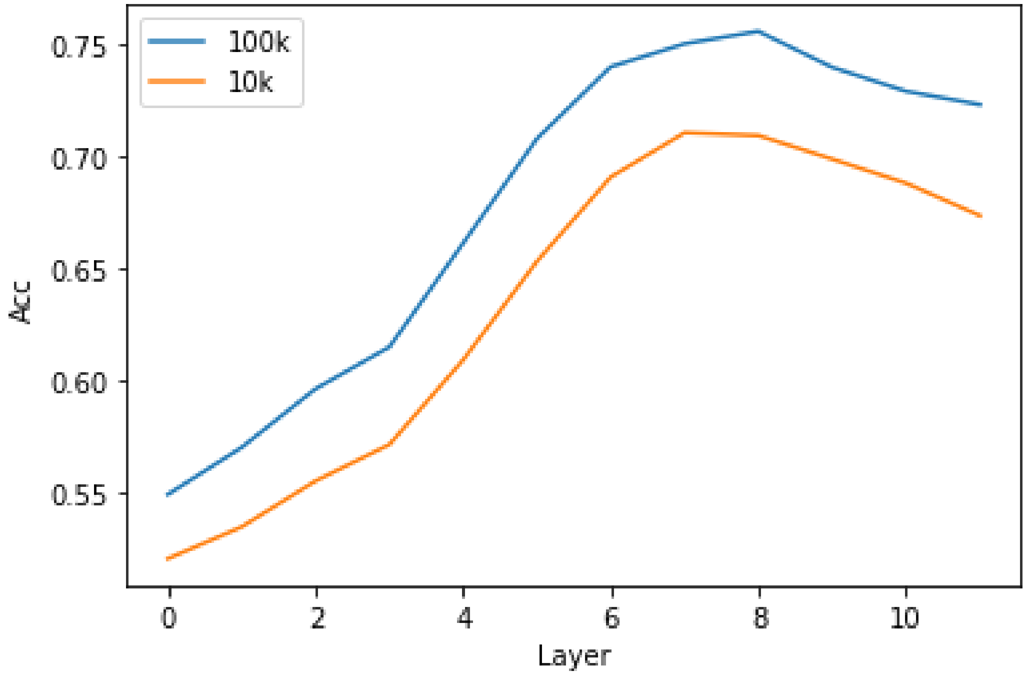
1. The first layers representation for tense is “##ed”, ”##s”, ”at”, ”for”
2. The 4 layers of representation for tense is “##ed”, ”##s”, ”was”, ”i”. This layer has no “##s” in its representations.
3. The 7 layers' representation for tense is “##s”, ”##ed”, ”i”, and ”she”. This layer has no “##s” in its representations either.
4. The 10 layers representation for tense is “-”, ”##s”, ”##ed”, ”i”.
5. The 12 layers representation for tense is “##S”, ”i”, ”[sep]”, ”##ed”.

Understandably, the first layers are better for this probing task.

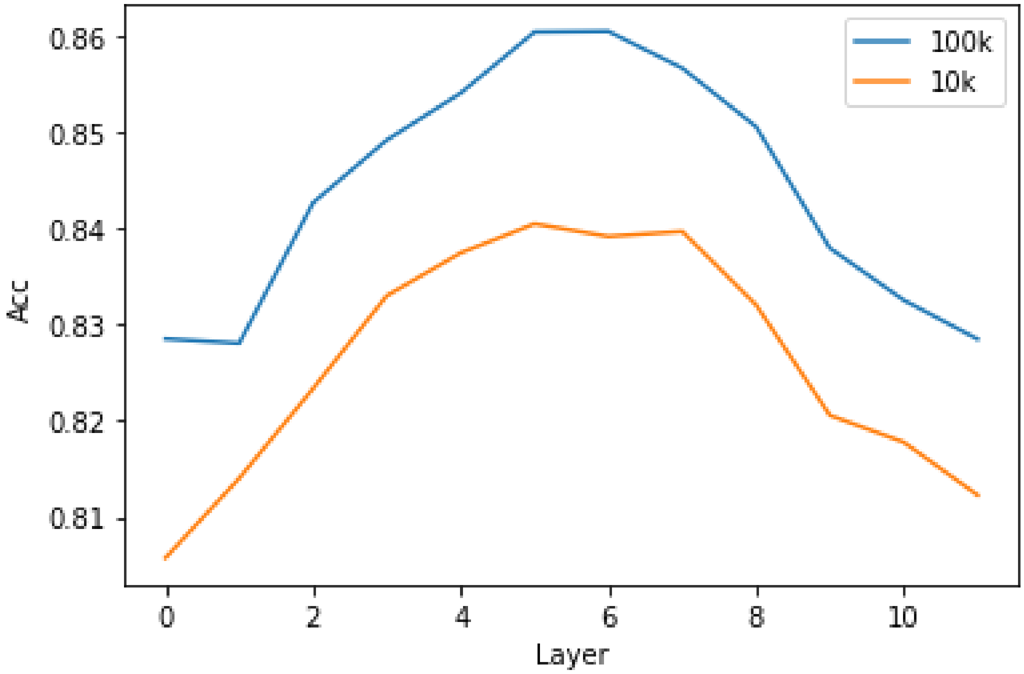
In control experiments the accuracy of each task for each layers representation is measured, as below:



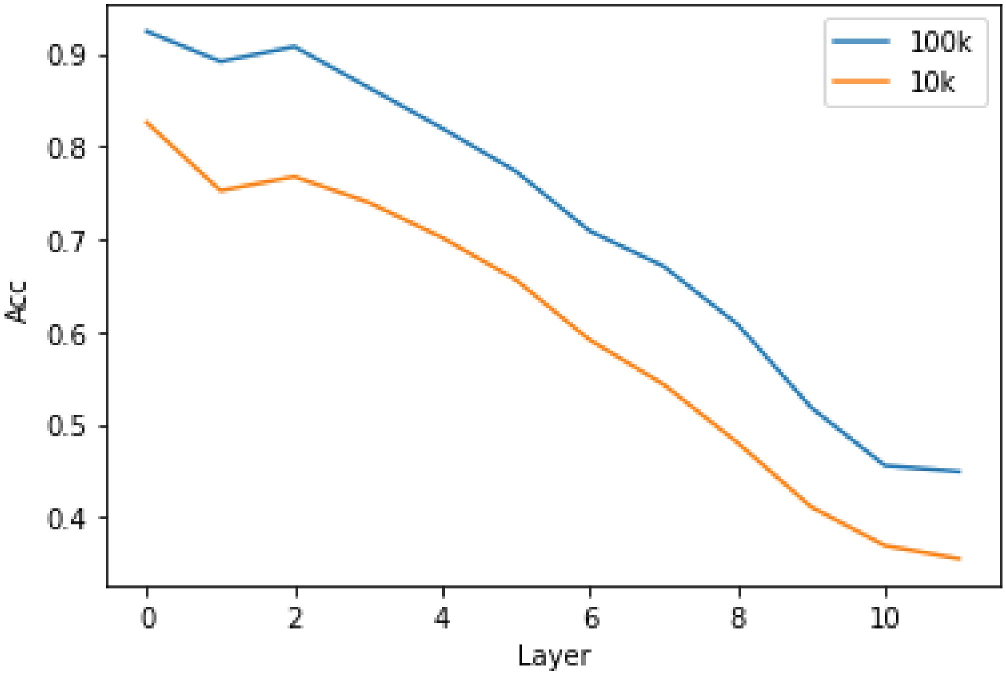
bShift



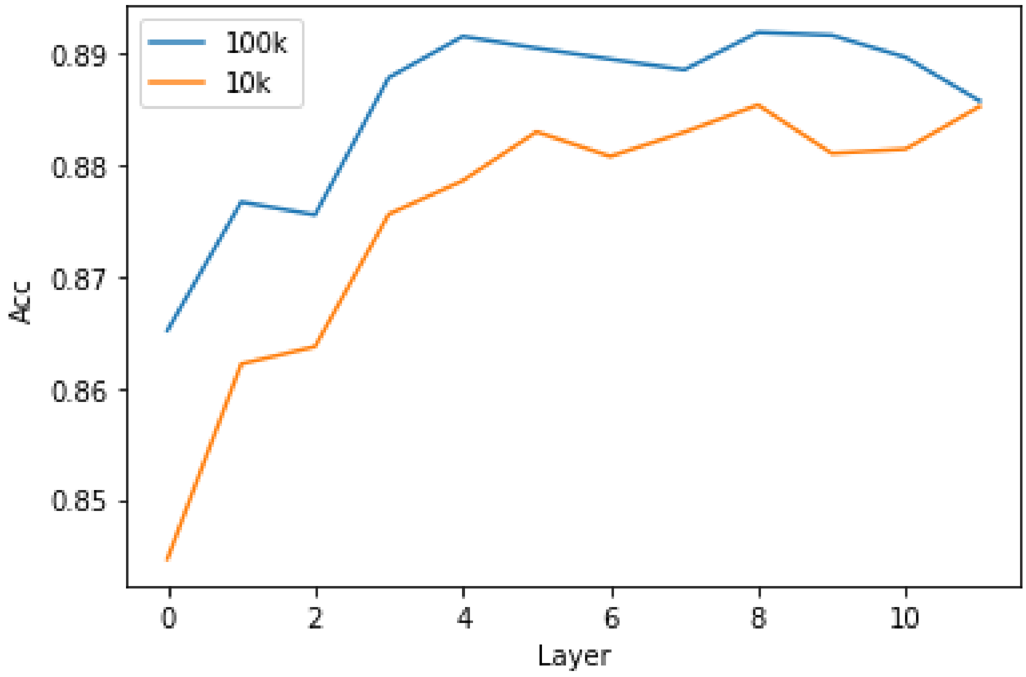
Coord



Objnum



Sentlen



Tense

Conception in final words

Bert is a very good option for probing tasks, and it has different performances for these tasks on its different layers.

The best layers representation for objnum are middle ones, and the poorest performances for these tasks were lst and first layers.

The best layers representation for tense were the last ones and the poorest were the first layers. And this is in exact opposite for Sentlen.

Bert is not so powerful in the Bshift task.

The best layers representation for the coord task are for 8 and 9.