How to verify if embedding is of good quality for health care related data?

I am going to discuss 3 approaches to very embedding quality:

1st: cosine similarity between medical keywords pair:
We have two files ClinNotes.csv and MedicalConcepts.csv and we can utilize them to check embedding quality. First approach is to check cosine similarity between pairs of keywords. If the two pairs are equal(synonym) then cosine similarity is 1, if the pairs are opposite(antonym) then it is -1, and if the keyword pair is completely unrelated then the cosine similarity is 0. I will calculate the cosine similarity for each pair(we have 558 unique pairs) in the MedicalConcepts.csv file and check if the mean cosine similarity is near to 1 or not. A good model which can understand health related data should give mean cosine similarity near to 1.

2nd approach: generate embedding for ClinNotes.csv data and draw a scatter plot:
I will generate embedding for ClinNotes.csv data and draw a scatter plot for embedding against the note's category column. If the embeddings are of good quality, all the cardiovascular related vectors should be closer in the plot, all the Neurology related vectors should form another group and the same goes for the 3rd category. For creating a scatter plot, I am first transforming embeddings in 2 dimensions using T-SNE and then plotting it.

3rd: approach: perform K-means clustering (in our case k is equal to 3 as there are 3 categories of clinic notes)
perform clustering on embeddings obtained from clinic notes and then check which type of category's data produces the similar type of embedding. In other ways we want to check which model is getting confused among different types of category's data.

To find embeddings for health related data i have used below techniques-

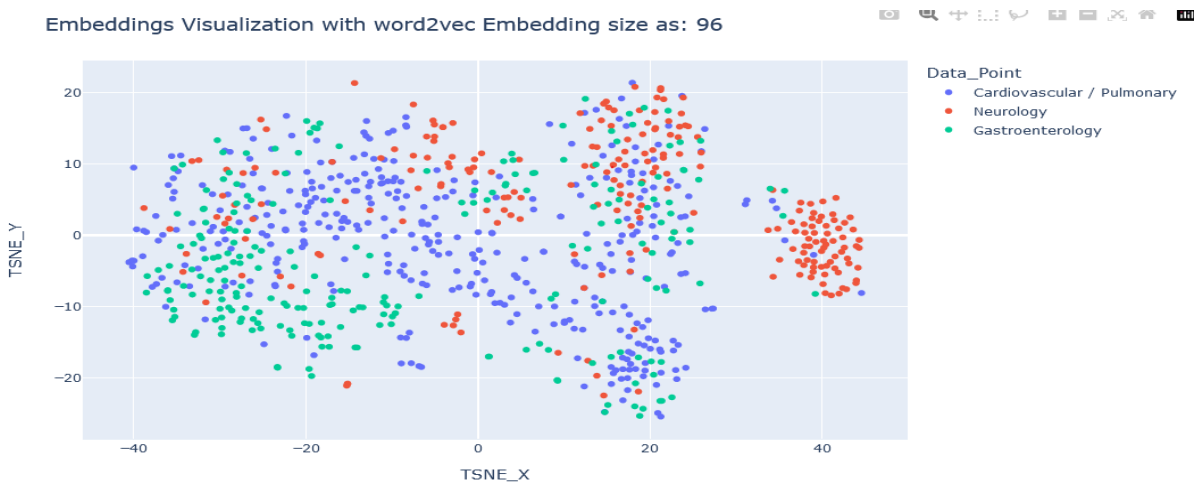
1st: Word2Vec
I have started off with the simplest semantic similarity approach based embedding technique: word2vec. I have used spacy for getting word2vec embedding. Sapcy has multiple models using which word2vec embeddings can be obtained and these model are already trained on huge corpus of data. I have used en_core_web_sm and en_core_web_lg models. en_core_web_sm is a small model and gives 96 length long embedding and en_core_web_lg is a larger model which gives 300 word long embedding.

Though I have used Word2vec for comparison purposes, it has a major problem and that is, it can not identify the context in which the word appears in a sentence. Second problem is that this model is trained on generic corpus and might not give good embedding for health related keywords.

Below is the result of cosine similarity on medical keyword pairs using word2vec small model.

```
Total number of pairs =   558
Total cosine_similarity =   239.9576534051448
mean cosine_similarity =   0.43003153800964355
```

Note the last value “mean cosine similarity = 0.43”, which is not very good. we want similarity closer to 1.
Now below is the result obtained when I fed ClinNotes.csv data in word2vec model and plot it against category column.



No clear group can be seen in the above plot and this proves that embeddings are not good.

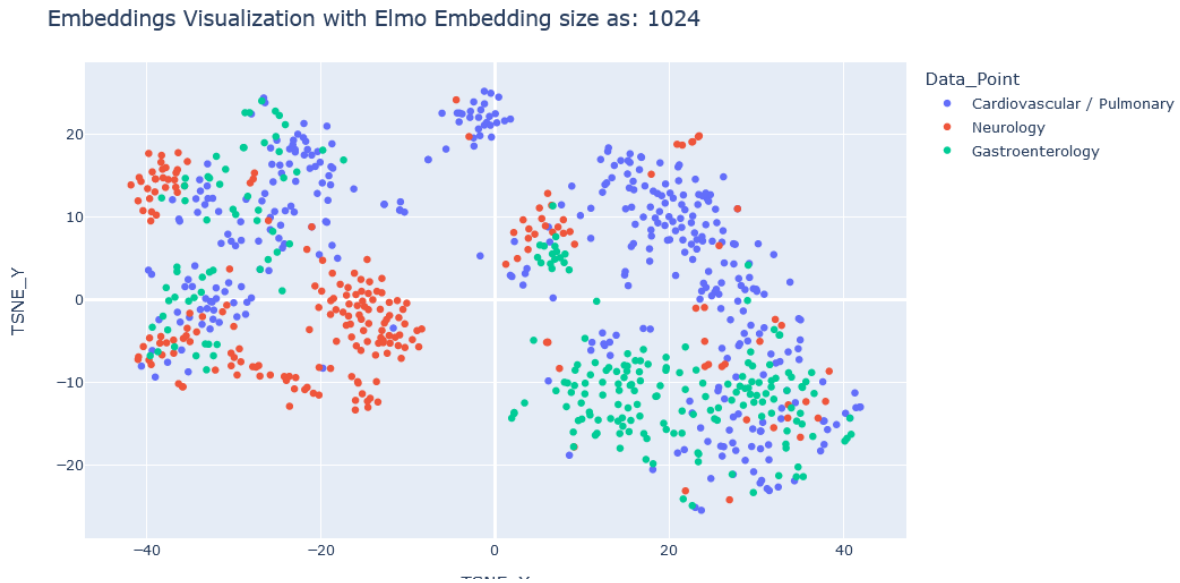
Now let's check where model is getting confused by utilising k-means clustering result-

```
K-Means clustering summary on Word2Vec 96
**********Data points in cluster: 0  :**************
Neurology                     158
Cardiovascular / Pulmonary    122
Gastroenterology               80
Name: category, dtype: int64
**********Data points in cluster: 1  :**************
Cardiovascular / Pulmonary    195
Gastroenterology              123
Neurology                      54
Name: category, dtype: int64
**********Data points in cluster: 2  :**************
Cardiovascular / Pulmonary    54
Gastroenterology              21
Neurology                     11
Name: category, dtype: int64
```

Cluster 0 is getting confused greatly between neurology and cardiovascular by assigning 158 neurology points and 122 cardiovascular points to cluster 0. Same goes for cluster 1 as well in which model is confusing between cardiovascular and gastroenterology. This of course is not a good result.

2nd: ELMO

Second technique was to get embedding from the Elmo model. This model is better than word2vec and can capture the context of a sentence pretty well. But again this model too is not specifically trained on health related data and might not give good embedding for health related keywords. Below is the result when i fed ClinNotes.csv data to ELMO and plot obtained vectors against category



I think the results are far better than the word2vec. We can see better groups in data. Let's check cosine similarity on keyword pair

```
Total number of pairs =   558
Total cosine_similarity =   308.23337239027023
mean cosine_similarity =   0.5523895
```

The mean cosine similarity is 0.55, which is not too bad. As we saw above as well, ELMO did perform better than word2vec.

let's check k-means clustering result-

```
K-Means clustering summary on Elmo
***********Data points in cluster: 0  :***************
Cardiovascular / Pulmonary    135
Neurology                      58
Gastroenterology               28
Name: category, dtype: int64
***********Data points in cluster: 1  :***************
Neurology                     141
Cardiovascular / Pulmonary    113
Gastroenterology               70
Name: category, dtype: int64
***********Data points in cluster: 2  :***************
Gastroenterology              126
Cardiovascular / Pulmonary    123
Neurology                      24
Name: category, dtype: int64
```

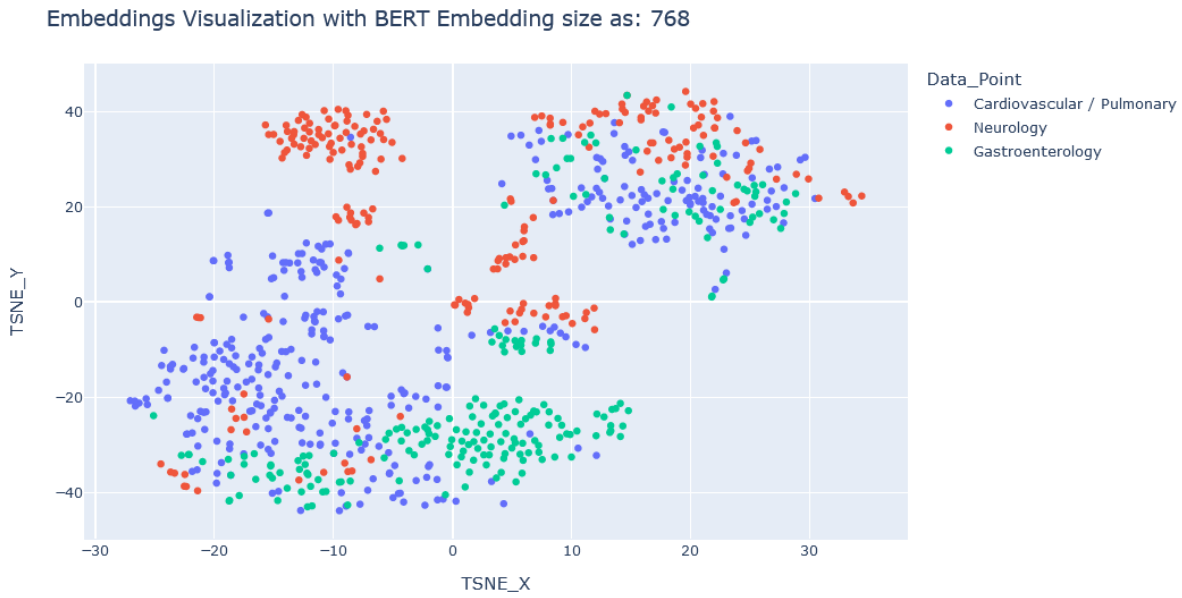
K-Means shows that ELMO embeddings too are not of good quality. It is getting confused greatly in cluster 2 and in other clusters too there is some confusion.

3rd: BERT

Third model that I tried was BERT which is a transformer based model and can capture context of the sentence pretty well. There are many variations of the BERT model, few of them are specifically trained on health related data. I have first started off with the BERT base uncased model which is trained on generic corpus. Below is the result of cosine similarity on keyword pairs

```
Total number of pairs =   558
Total cosine_similarity =   319.710486933589
mean cosine_similarity =   0.57295823097229
```

BERT base model has performed even better than ELMO on keyword similarity task, Though this cosine similarity(0.57) too is not very good.Now lets plot clinic notes using bert embeddings



We can see the groups getting formed in the above plot, but again they are not very good.

Lets see k_means clustering result-

```
K-Means clustering summary on Bert Base
***********Data points in cluster: 0  :**************
Cardiovascular / Pulmonary    78
Neurology                     53
Gastroenterology              24
Name: category, dtype: int64
***********Data points in cluster: 1  :**************
Neurology                    145
Cardiovascular / Pulmonary   120
Gastroenterology              56
Name: category, dtype: int64
***********Data points in cluster: 2  :**************
Cardiovascular / Pulmonary   173
Gastroenterology             144
Neurology                     25
Name: category, dtype: int64
```

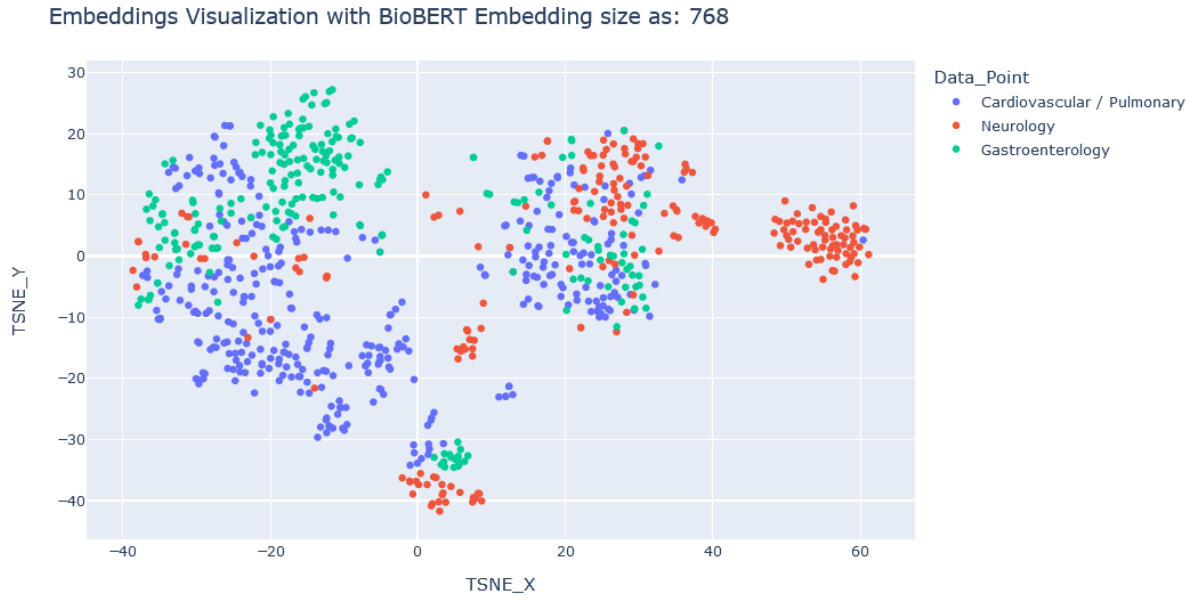
Again K-means clustering shows that embeddings are not of good quality

4th: BioBERT

Now I have used another variant of BERT called BioBERT. This model is trained on a huge corpus of biomedical literature. I believe this model should be able to perform better than any other model that we have used so far. Below is the cosine similarity result from this model-

```
Total number of pairs =   558
Total cosine_similarity =   447.63191509246826
mean cosine_similarity =   0.8022072911262512
```

As you can see, mean cosine similarity is 0.80, which is way higher than any other model we used so far. Let's plot clinic notes data with this model's embedding



This plot looks little better than the BERT base model's plot.

Lets see k-means result-

```
K-Means clustering summary on BioBert
***********Data points in cluster: 0  :*************
Neurology                      151
Cardiovascular / Pulmonary     106
Gastroenterology                53
Name: category, dtype: int64
***********Data points in cluster: 1  :*************
Cardiovascular / Pulmonary     156
Gastroenterology               146
Neurology                       25
Name: category, dtype: int64
***********Data points in cluster: 2  :*************
Cardiovascular / Pulmonary     109
Neurology                       47
Gastroenterology                25
Name: category, dtype: int64
```

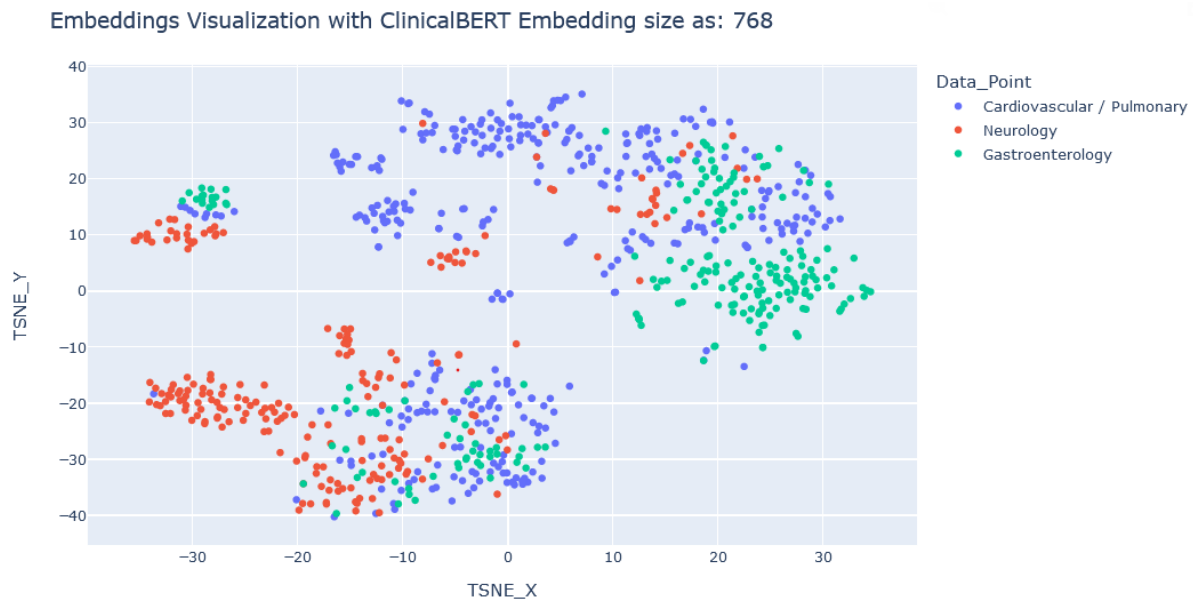
Again k-means shows that produced embeddings are confusing.

5th: ClinicalBERT

Another variant of BERT is ClinicalBERT. This model is trained in a huge corpus of clinic notes data, which can be advantageous as we too are dealing with clinic notes data. Below is the cosine similarity result from this model-

```
Total number of pairs =   558
Total cosine_similarity =   453.973201751709
mean cosine_similarity =   0.8135718107223511
```

ClinicalBERT has given even better results than BioBERT on keyword pair similarity task. Lets plot ClinNotes data with model's embedding



This plot is even cleaner than BioBERT.

Lets see k-means result with this model-

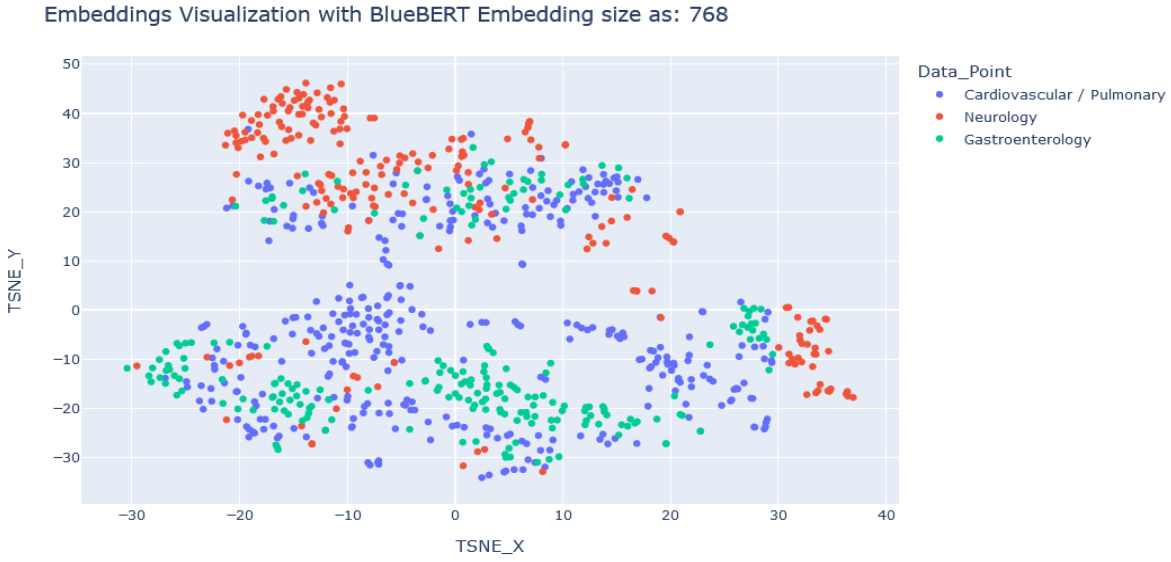
```
K-Means clustering summary on ClinicalBERT
***********Data points in cluster: 0  :***************
Cardiovascular / Pulmonary    82
Neurology                     51
Gastroenterology              27
Name: category, dtype: int64
***********Data points in cluster: 1  :***************
Neurology                     146
Cardiovascular / Pulmonary    106
Gastroenterology               50
Name: category, dtype: int64
***********Data points in cluster: 2  :***************
Cardiovascular / Pulmonary    183
Gastroenterology              147
Neurology                      26
Name: category, dtype: int64
```

K-Means shows that clinical bert too is producing ambiguous embeddings for clinic notes data.

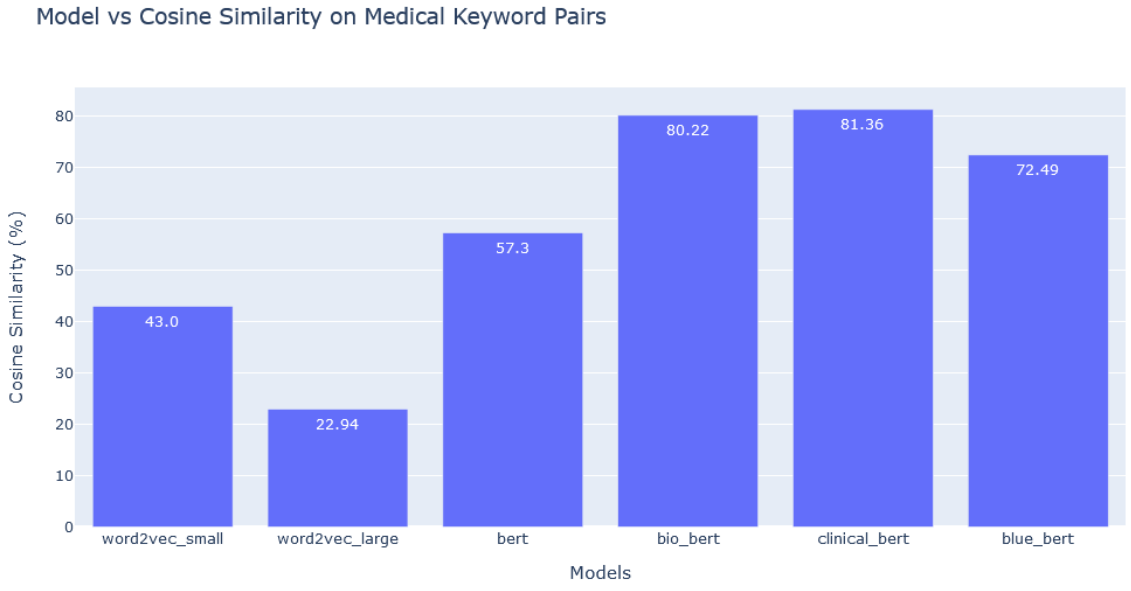
6th BlueBERT:
This is another variation of BERT, which again trained on biomedical and clinical notes. Let's check the cosine similarity obtained from this model-

```
Total number of pairs =   558
Total cosine_similarity =   404.4722881615162
mean cosine_similarity =   0.7248604893684387
```

This model is not as good as BioBert and ClinicalBERT. Lets also verify the embedding performance on ClincNotes data as well-

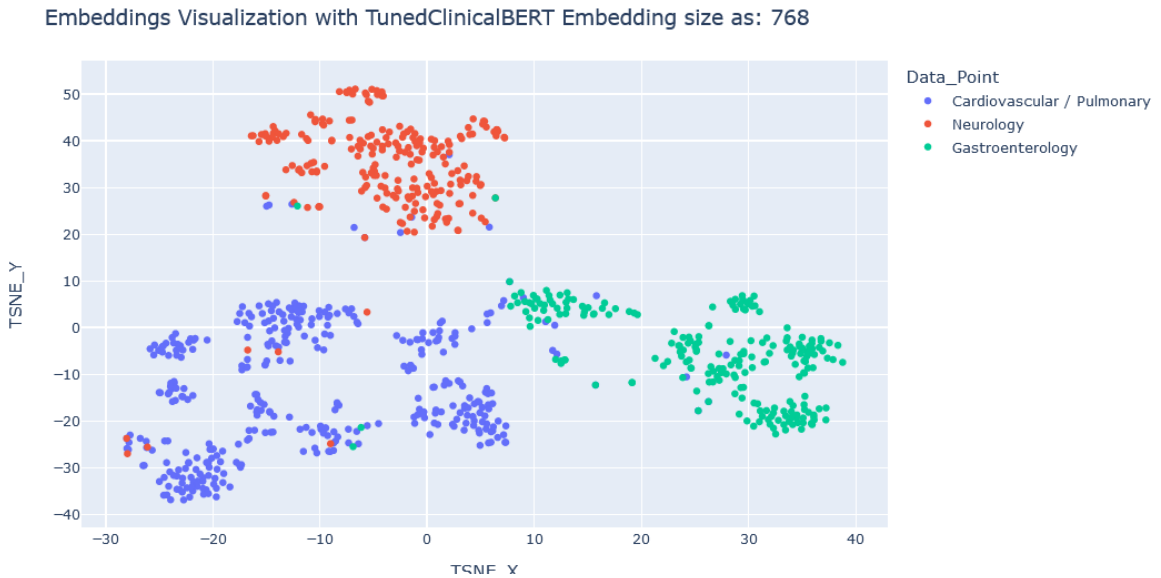


Scatter plot result too is worse than BioBERT and ClinicalBERT.
Now Let's check all models' performance together on a cosine similarity task..



ClinicalBERT is best with 81.36% similarity score on keyword pair. Its scatter plot too is better than any other plot we saw so far-

Because Clinical BERT seemed to be performing better, I thought of fine tuning it even further. I trained it on ClinNotes.csv data to predict the category given notes to it. It was a 3 class classification problem. Before feeding it into the model, I performed some preprocessing steps such as removing special symbols and multiple spaces and dividing lengthy notes in smaller chunks. New chunks created had new rows with the same category. Note that chunking was important, otherwise we might have lost important information while training with BERT as it can process only 512 tokens in one go, the rest of the tokens would have been ignored. Once this model was trained, i used it to get embeddings for clinic notes data and mapped them against category below is the result i got-



This fine tuned model has formed the best groups and the reason is obvious, i have trained it on similar kinds of data. With this model I noticed that the similarity score has reduced a little. Below is the similarity score on keyword pair.

```
Total number of pairs =   558
Total cosine_similarity =   433.7339897155762
mean cosine_similarity =   0.7773011922836304
```

Lets check K-Mean analysis for this model-

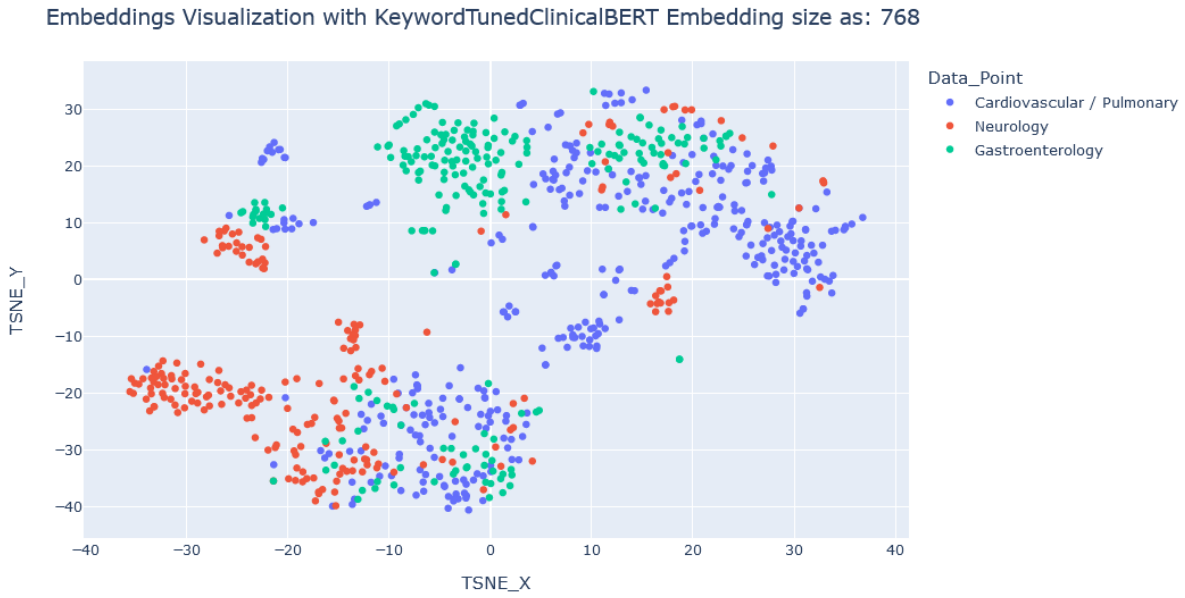
```
K-Means clustering summary on tuned_clinical_bert
***********Data points in cluster: 0  :**************
Gastroenterology              219
Cardiovascular / Pulmonary     49
Name: category, dtype: int64
***********Data points in cluster: 1  :**************
Neurology                     216
Cardiovascular / Pulmonary     10
Gastroenterology                4
Name: category, dtype: int64
***********Data points in cluster: 2  :**************
Cardiovascular / Pulmonary    312
Neurology                       7
Gastroenterology                1
Name: category, dtype: int64
```

K-means shows no confusion in clustering embeddings from this model. This indeed has produced the best embedding for clinic notes and the reason is obvious that model is finetuned using the same type of data

I also tried improving the performance on keyword pair similarity tasks, for which I tried the below technique.
Note that for this task too I am fine tuning the pre trained ClinicalBERT model.
:
For each right keyword pair , I created and wrong keyword pair. Then I mapped the right keyword pair with 1 and wrong keyword pair with 0. Now it became a binary classification problem where the task was to identify if a pair is equal to 1 or 0. I trained this model just for 1 epoch and I found a slight improvement in cosine similarity. Below is the result-

```
Total number of pairs =   558
Total cosine_similarity =   456.11888909339905
mean cosine_similarity =   0.8174178600311279
```

As you can see, mean cosine similarity is 0.8174. Note that, with the pretrained ClicalBert model, it was 0.8135. I know it's a slight improvement (I just ran it for 1 epoch), but I think results can further be improved with right hyperparameter tuning and more data. I could not perform hyperparameter tuning because of insufficient computation power.
Now let's plot ClinicNotes data embeddings from this model.



Formed groups are actually not too bad, but also not very good and the mean cosine similarity score too is the highest so far.

Lastly, I trained ClinicalBERT on clinic notes data first and then on keyword pair and below is the result obtained on cosine similarity task.

```
Total number of pairs =   558
Total cosine_similarity =   436.2237065434456
mean cosine_similarity =   0.78176296
```

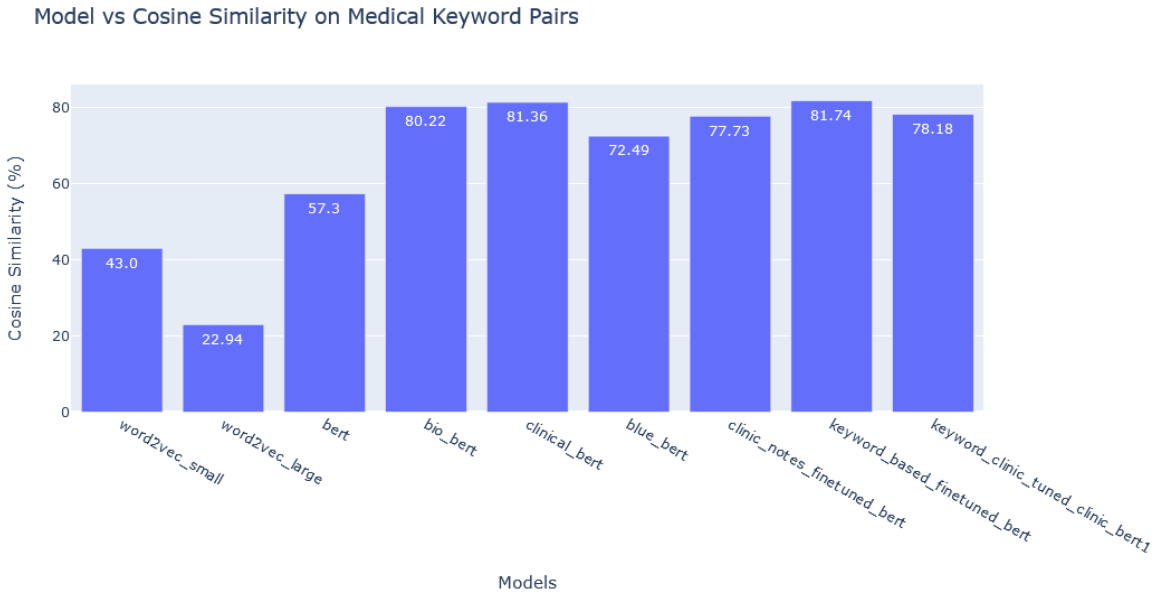
Cosine similarity(0.7817) has improved a little when it is compared to the model that was solely trained on clinic notes. That model had a similarity score as 0.7773. As stated earlier as well, this similarity score can further be increased with more data. Now let's check the cluster formation as well-

Clusters are very well organised. This has got some improvement for sure.

```
K-Means clustering summary on NotesAndKeywordTunedClinicalBERT
**********Data points in cluster: 0  :**************
Neurology                      146
Cardiovascular / Pulmonary     106
Gastroenterology                50
Name: category, dtype: int64
**********Data points in cluster: 1  :**************
Cardiovascular / Pulmonary     82
Neurology                      51
Gastroenterology               27
Name: category, dtype: int64
**********Data points in cluster: 2  :**************
Cardiovascular / Pulmonary     183
Gastroenterology               147
Neurology                       26
Name: category, dtype: int64
```

Again result from this model to seems confusing only when analysed using K-means

Now let's verify each models performance on keyword cosine similarity task together-



Keyword_based_fine_tuned_bert and given the best similarity score, but we should not just focus on keyword similarity score. As per my analysis so far, Pertained ClinicalBERT model, ClinicalBERT tuned on keyword pair task and ClinicalBERT tuned on clinic notes and keyword pair both could be researched even further.

Conclusion:

1. K-means might now have given good results with most of the models, but it has certainly guided us to check the data for categories where models are getting confused.

2. Looking at the result of all three approaches of analysing the embedding quality, I believe that the ClinicBERT model, which was trained on clinic data only, has produced very decent results. Its cosine similarity was 0.77. It produced unambiguous embeddings when analysed through K-Means

Other suggestions:
One other approach that we can try is checking keywords in Clinic Notes that do not occur together, which mean either Term1 occurs or Term2 occurs. We then place the pair of it in the sentence and train the model for notes category classification task. By doing so, embedding of keywords appearing in single note will come closer.