COURSEWORK 2 REPORT

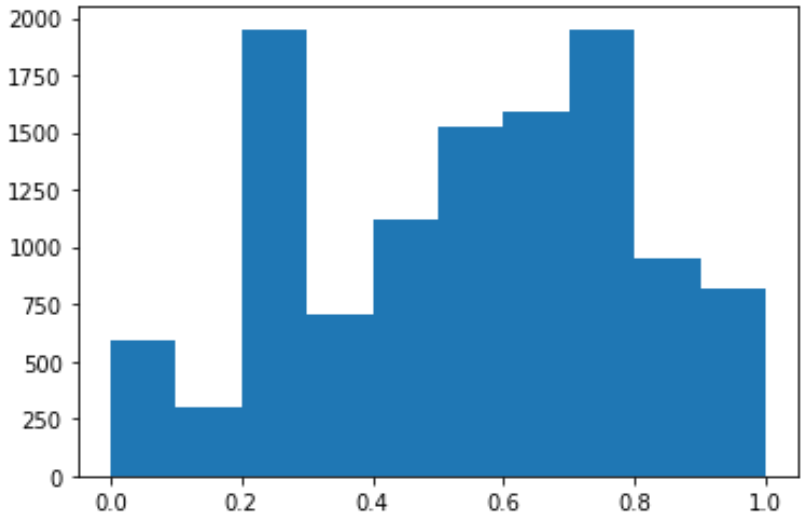
Data Analysis- The dataset contains pairs of sentences and their similarity score ranging from 0 to 1 where 1 corresponds to maximum semantic similarity and 0 corresponds to minimum semantic similarity. The training set has 11498 examples and developer set consists of 3000 examples.

	Unnamed: 0	Sent1	Sent2	SimScore
0	0	U.S., EU Widen Sanctions On Russia	U.S., EU Boost Sanctions On Russia	1.00
1	1	The lawyers advised the judges .	The lawyers advised the judges behind the acto...	0.79
2	2	Man kills 4 in Calif. before police shoot him ...	Police: Gunman killed 6 in California shootings	0.40
3	3	Someone is playing a piano.	A man is playing a guitar.	0.24
4	4	In an E-mail statement to the Knoxville News S...	I am not giving any consideration to resignati...	0.80
5	5	The secretaries supported the managers present...	The managers presented in the library .	0.59
6	6	The author saw the manager .	Before the lawyer helped the banker , the auth...	0.78
7	7	The student and the senator supported the doct...	The senator supported the student .	0.29
8	8	The managers contacted the students paid in th...	The students paid in the library .	0.59
9	9	A man is spinning.	A man is dancing.	0.32
10	10	china is an important force for safeguarding w...	china is an important force for promoting worl...	0.68
11	11	The tourist next to the professors shouted .	The professors shouted .	0.26
12	12	Two green and white trains sitting on the tracks.	Two green and white trains on tracks.	0.88
13	13	The banker arrived .	Of course the banker arrived .	0.88
14	14	China sets Bo Xilai trial date	China Bo Xilai trial in fourth day	0.64
15	15	Judge Leroy Millette Jr. can reduce the punish...	Though the judge can reduce the punishment to ...	0.65
16	16	AOL says to sell 800 patents to Microsoft for ...	AOL to sell 800 patents to Microsoft for $1 bi...	0.96
17	17	A woman is picking tomatoes.	A woman is pouring batter into a bowl.	0.20
18	18	The tourist paid the author .	Although the tourist paid the author mentioned...	0.23
19	19	The managers helped the senator .	The managers that believed the tourist helped ...	0.79

In the examples, zeroth row has the same meaning but uses ‘Boost’ instead of ‘Widen’, and therefore has the maximum semantic similarity. Similarly, 9th row has low semantic similarity due to the difference between the meaning of ‘Spinning’ and ‘Dancing’.

The text appears to be clean and doesn’t require any extra preprocessing. Removal of stop words was tried on a subset, but it didn’t make the performance of our model better.

The following histogram shows the distribution of semantic similarity in our training set. It shows there are only a few examples with very low semantic similarity.



Features/ Embedding selection- To provide a better description of every word, I have preferred 300 dimensional embeddings. Earlier whilst developing the model, I started with Glove6B300 embedding and tested my model on that. Later, I tried performance of several embedding, including Word2Vec google news, Glove840B300 which is the biggest embedding I tried. The performance of these embedding didn't make much impact on the Train MSE or Dev/Test MSE.

However, Glove6B300 worked slightly better than the others, which I assume that's because it was trained on more tokens. Another reason for me to choose this embedding was the histogram plot of absolute and squared errors. The performance seemed better on histogram plots, as using this embedding reduced the magnitude of difference between the predicted similarity and true similarity I.e., there are less instances when our predicted similarity is different than true similarity by, say, 0.6 or higher. Therefore, I have chosen Glove6B embedding.

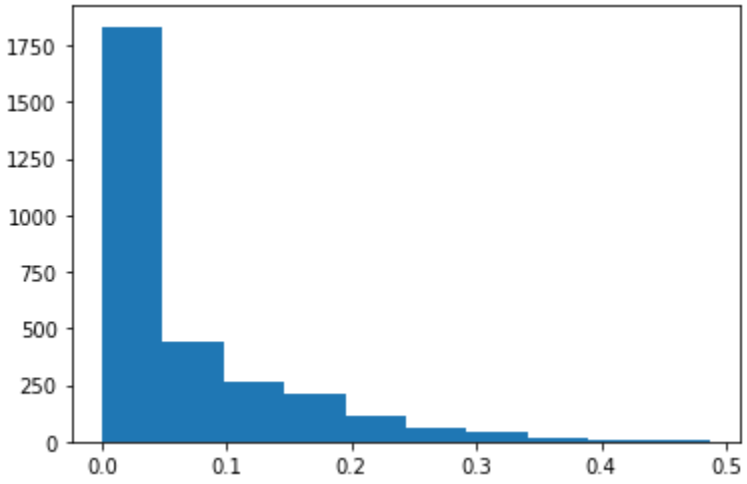
TASK 1- MLP-based encoder

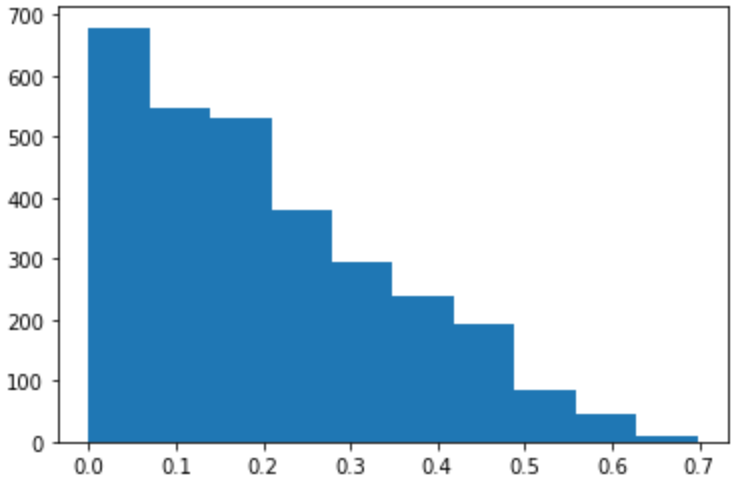
Selecting the architecture- I first started selecting my architecture by starting with just one layer with 300 neurons and test its performance, after training I saw MSE was about 12%. I clearly felt the need to add more layers. It was a trial and error; I followed the strategy of checking the MSE error on the training set and dev/test set continuously to see the effect each update has on them. MLP with 2 hidden layers with 300 neurons each without dropout gave good results. Then, I tried 3 and 4 layered architectures. The 4 layered architecture was a clear case of overfitting and despite the use of heavy dropout the performance was better on 2 and 3 layered architectures. For, the three-layered architecture, I changed the size of each layer from

0.66 of the previous layer to 1.25 of the previous layer. After validating through many architectures, I found the 3 layered architecture with 0.2 dropout at the first layer with 400 neurons, followed by no-dropout at the second layer of 350 neurons and the third layer with 350 neurons. For the sentence representation, I tried Average embedding and concatenation of word embeddings, whilst there was no major difference between the two representations, I have chosen the average embeddings representation based on histogram plot obtained.

Other hyperparameters- I chose the learning late = 1e-4. I tried several learning rate and 1e-4 turned out to be the optimal rate. Slightly Increasing/decreasing the learning rate made the model to converge slower. I chose it via grid search. For others hyperparameters, learning rate decay of 0.85 worked the best for my model as It helped the model converge faster. I chose the batch size of 64 and it worked better than the batch size of 32 and 128. I proceeded with ADAM optimiser which worked well in this case and I didn't need to experiment with too many other optimisers. I trained the model for 7 epochs.

Error-Analysis- MSE is used as the loss function and optimisation is done using it. MSE of the method on the dev set is 0.061450418. The below histograms show the Error Analysis in the for of squared error and absoluter error respectively. For analysing, I have found absolute error more convenient. It can be seen in the absolute error histogram that the magnitude of absolute error is linearly decreasing. However, this is not a very good model as there are plenty of examples where our prediction is off by more than 0.3. However, this behaviour is not very surprising, since the MLP model is not very suitable for making the sentence embedding due to lack of ability to provide better sentence representation. The RNN/CNN models will help solve this problem.





Below is a list of 20 examples where our prediction was most off. The List is not ordered but gives an account of the kind of error. **Predicted_Sim in this table does not represent the predicted sim but represents the difference between Predicted_sim and SimScore.** (I noticed it a bit late to change the screenshot). The table shows that there are cases where the SimScore is on either extreme, or our model does a very poor job at predicting similarity. Many of the examples seem to have nothing in common at all. Using models that consider sequence and have better sentence representation as input are likely to provide a much better result.

	Unnamed: 0	Sent1	Sent2	SimScore	Predicted sim
951	951	Reading your post, made me think of my younger...	I'd say you need to distract yourself from the...	0.00	0.600472
569	569	They are preparing for a performance at school.	Two medical professionals in green look on at ...	0.00	0.604026
850	850	People who are good at the philosophy of mathe...	My motivation is that I like to look at things...	0.00	0.604073
927	927	I was invited in to pitch story ideas to Ron M...	It's obviously possible for it to leave the fi...	0.00	0.604897
84	84	The man used a sword to slice a plastic bottle.	A man sliced a plastic bottle with a sword.	1.00	0.394783
903	903	The important thing I try to remember is just ...	I have been reading on this topic since I have...	0.00	0.634891
638	638	I agree with Seteropere completely, "Network S...	I would say you are approaching it in the wron...	0.00	0.608686
359	359	Depressed woman sitting on couch.	Older woman holding newborn baby.	0.00	0.683123
665	665	There are individuals who possess extraordinar...	There are many arguments for why this is not t...	0.00	0.614101
859	859	The question of time is an incredibly difficul...	wikipedia, as much as it is a supposedly self ...	0.00	0.647173
1453	1453	Manchester United Ticket Sales And Profit Down	Manchester United owner Malcolm Glazer dies at 85	0.12	0.750880
17	17	A man is climbing a rope.	A man climbs a rope.	1.00	0.331650
728	728	First of all, as a general matter of fact you ...	You can mix those, but in my experience, it wi...	0.00	0.616434
957	957	Actually, it's much more easier to count the o...	That is also the recommended strategy for mara...	0.00	0.697981
238	238	A man plays the guitar and sings.	A man is singing and playing a guitar.	1.00	0.372725
960	960	I can't think of the specifics, but I seem to ...	The Laws of Cricket say that you can declare a...	0.00	0.653590
970	970	I had a similar issue but in what seemed to be...	I'm actually about to do the same thing when t...	0.00	0.632360
153	153	The man is buttering the bread.	The man is stirring the rice.	0.08	0.686825
973	973	There's a geek answer to this, and a practical...	It's pretty ridiculous that I've seen airlines...	0.00	0.646991
908	908	Even though the question has been answered I w...	I don't know if I should say this, but if your...	0.00	0.638349

TASK 2- RNN-based encoder

For the Task 2, I have chosen the RNN-based encoder. I chose RNN over CNN, because RNN considers the sequence of words and is a sequential model, and in sentence encoding works better than CNN in theory. For, this task too, I have used Glove840B300 embedding, and the reason and results which led to choosing it remains the same as in the previous section.

Selecting the architecture- I first started selecting my architecture by starting with just one hidden layer with 300 neurons and using the normal RNN architecture, which whilst performing better than MLP was still not the best. The next thing I tried was selecting Bidirectional LTSM based RNN architecture with one hidden layer as advised in the lecture. It performed reasonably well. I tried adding another RNN layer, but it showed signs of overfitting. I then added dropout varying from 0.2 to 0.5. However, even with dropout, the performance didn't match the expectations set by one hidden layer. I tried changing the second RNN layer to a fully connected layer as an experiment, but the results were very poor and discouraged further use of fully connected layers in RNN network for our dataset. For pooling, I tried Average pooling and Max pooling. Max pooling turned out to be a better bet for our dataset.

I also tried Bidirectional GRU, and unidirectional version of both GRU, LSTM. But as expected, bidirectional counterparts did a better job. In terms of MSE, the difference between the performance of Bidirectional GRU was as good as Bidirectional LSTM, but Bidirectional LSTM being better theoretically performed marginally better and hence, I chose bidirectional LSTM.

MSE of the best architecture on the dev set is **0.022532336.**

Other hyperparameters- I chose the learning late = 0.005. I tried several learning rates and 0.005 turned out to be the optimal rate. Slightly decreasing the learning rate made the model to converge slower, whilst increasing it showed divergence after some training. I chose it via grid search. For others hyperparameters, learning rate decay of 0.999 worked the best for my model as It helped the model converge faster. I chose the batch size of 64 and it worked better than the batch size of 32 and 128. I proceeded with ADAM optimiser which worked well in this case. I used gradient clipping with clipping value = 3. I trained it for 7 epochs.

Below is training time values and loss on train and dev/test set.

0%| | 0/7 [00:00<?, ?it/s]

tensor(0.1835, device='cuda:0', grad_fn=<MseLossBackward>)tensor(0.0593, device='cuda:0', grad_fn=<MseLossBackward>)tensor(0.0341, device='cuda:0', grad_fn=<MseLossBackward>)tensor(0.0318, device='cuda:0', grad_fn=<MseLossBackward>)======epoch 0 loss====== 0.050923757

14%|■▌ | 1/7 [02:39<15:57, 159.52s/it]

---> after epoch 0 the MSE on dev set is 0.03730253502726555learning rate 0.005best model updated; new best MSE tensor(0.0373, device='cuda:0')tensor(0.0285, device='cuda:0', grad_fn=<MseLossBackward>)tensor(0.0326, device='cuda:0', grad_fn=<MseLossBackward>)tensor(0.0214, device='cuda:0', grad_fn=<MseLossBackward>)tensor(0.0156, device='cuda:0', grad_fn=<MseLossBackward>)======epoch 1 loss====== 0.025448386

29%|■■▌ | 2/7 [05:18<13:16, 159.37s/it]

---> after epoch 1 the MSE on dev set is 0.030031252652406693learning rate 0.004995best model updated; new best MSE tensor(0.0300, device='cuda:0')tensor(0.0174, device='cuda:0', grad_fn=<MseLossBackward>)tensor(0.0166, device='cuda:0', grad_fn=<MseLossBackward>)tensor(0.0122, device='cuda:0', grad_fn=<MseLossBackward>)tensor(0.0106, device='cuda:0', grad_fn=<MseLossBackward>)======epoch 2 loss====== 0.015718486

43%|■■■■▏ | 3/7 [07:57<10:36, 159.23s/it]

---> after epoch 2 the MSE on dev set is 0.026526402682065964learning rate 0.0049900050000000005best model updated; new best MSE tensor(0.0265, device='cuda:0')tensor(0.0123, device='cuda:0', grad_fn=<MseLossBackward>)tensor(0.0103, device='cuda:0', grad_fn=<MseLossBackward>)tensor(0.0105, device='cuda:0', grad_fn=<MseLossBackward>)tensor(0.0095, device='cuda:0', grad_fn=<MseLossBackward>)======epoch 3 loss====== 0.011289936

57%|■■■■■▍ | 4/7 [10:36<07:57, 159.27s/it]

---> after epoch 3 the MSE on dev set is 0.02421095222234726learning rate 0.004985014995best model updated; new best MSE tensor(0.0242, device='cuda:0')tensor(0.0081, device='cuda:0', grad_fn=<MseLossBackward>)tensor(0.0069, device='cuda:0', grad_fn=<MseLossBackward>)tensor(0.0071, device='cuda:0', grad_fn=<MseLossBackward>)tensor(0.0065, device='cuda:0', grad_fn=<MseLossBackward>)======epoch 4 loss====== 0.008861557

71%|■■■■■■▎ | 5/7 [13:15<05:18, 159.17s/it]

---> after epoch 4 the MSE on dev set is 0.023694319650530815learning rate 0.004980029980005best model updated; new best MSE tensor(0.0237, device='cuda:0')tensor(0.0083, device='cuda:0', grad_fn=<MseLossBackward>)tensor(0.0056, device='cuda:0', grad_fn=<MseLossBackward>)tensor(0.0062, device='cuda:0', grad_fn=<MseLossBackward>)tensor(0.0050, device='cuda:0', grad_fn=<MseLossBackward>)======epoch 5 loss====== 0.007432012

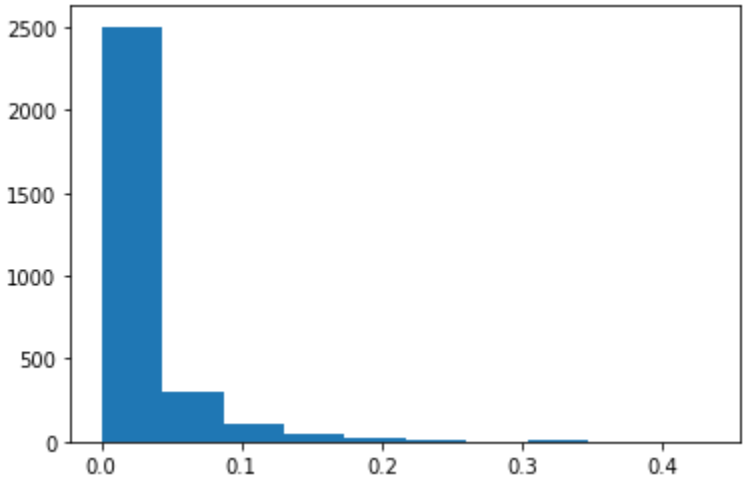
86%|■■■■■■■▊ | 6/7 [15:55<02:39, 159.23s/it]

---> after epoch 5 the MSE on dev set is 0.02251024916768074learning rate 0.004975049950024995best model updated; new best MSE tensor(0.0225, device='cuda:0')tensor(0.0056, device='cuda:0', grad_fn=<MseLossBackward>)tensor(0.0047, device='cuda:0', grad_fn=<MseLossBackward>)tensor(0.0066, device='cuda:0', grad_fn=<MseLossBackward>)tensor(0.0056, device='cuda:0', grad_fn=<MseLossBackward>)======epoch 6 loss====== 0.006539807

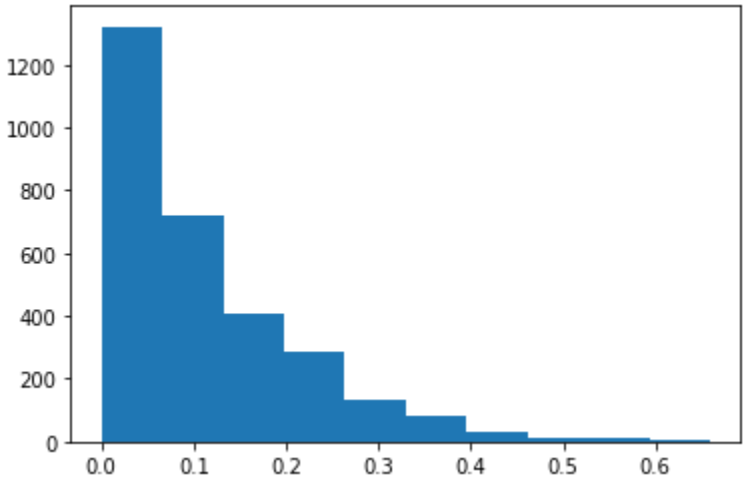
100%|■■■■■■■■■■| 7/7 [18:34<00:00, 159.19s/it]

---> after epoch 6 the MSE on dev set is 0.02253233641386032learning rate 0.00497007490007497

Error-Analysis- MSE is used as the loss function and optimisation is done using it. MSE of the method on the dev set is 0.02253233641386032. The below histograms show the Error Analysis in the for of squared error and absoluter error respectively. For analysing, I have found absolute error more convenient. It can be seen in the absolute error histogram that the magnitude of absolute error is exponentially decreasing unlike in the MLP where there was a linear decrease. It shows RNN's superiority over MLP for Sentence encoding tasks. This is a good model as there are only a few examples where our prediction is off by more than 0.3. However, this behaviour is not very surprising, since the RNN models are very suitable for these type of tasks due to their ability to use better sentence representation and taking sequential information into the account efficiently. The square error histogram is also a big improvement over the MLP, as most of the squared error are less than 0.1.



MSE of the method on the dev set: 0.061450418



Below is a list of 20 examples where our prediction was most off. The List is not ordered but gives an account of the kind of error. **Predicted_Sim in this table does not represent the predicted sim but represents the difference between Predicted_sim and SimScore.** (I noticed it a bit late to change the screenshot). The table shows a big improvement from the MLP scenario where the SimScore was on either extreme, or MLP model did a very poor job at predicting similarity. Unlike the MLP model, this model's error examples have more similarity than the last time. Some, mispredictions are arising due to change of numbers, day name, using short forms for states like Col. for Colorado.

Several methods can be used to make this model more efficient. Text preprocessing to change the numbers into words, changing the short forms to full forms. Using our own embeddings suitable for the task, or fine-tuning the pre-loaded embedding according to our task might provide better results.

Other methods, include using models like BERT or GPT to train the model.

	Unnamed: 0	Sent1	Sent2	SimScore	Predicted sim
714	714	In the United States, jails are operated by ci...	Jail is a municipal level, prison is on a stat...	0.72	0.242908
507	507	A man is carrying a canoe with a dog.	A dog is carrying a man in a canoe.	0.36	0.864232
510	510	The woman is drinking lemonade and watching T.V.	The man is sitting drinking coffee.	0.08	0.558704
1340	1340	Nelson Mandela taken to hospital	Nelson Mandela released from hospital	0.32	0.869790
1343	1343	Today in History, April 23	Today in History, Jan. 21	0.28	0.820144
1344	1344	UK alert on Syrian chemical arms	West raises stakes over Syria chemical claims	0.76	0.201413
492	492	A skateboarder jumps off the stairs.	A dog jumps off the stairs.	0.16	0.717765
1265	1265	Colorado shooting suspect was in therapy	Lawyers: Colo. shooting suspect is mentally ill	0.84	0.228916
767	767	The bulk of India then was not controlled by P...	I believe Alexander lost heart after his horse...	0.12	0.623804
1369	1369	Stocks close 0.39% higher	Stocks close 2.47% higher	0.36	0.945598
579	579	A man is throwing a penny into a fountain.	A little boy is throwing a man in water.	0.12	0.598147
1092	1092	BioReliance's stock closed down 2 cents yester...	Shares of BioReliance sold at $47.98 at the cl...	1.00	0.406720
617	617	The gate is blue.	The gate is yellow.	0.32	0.866974
981	981	I think the dual goals have a lot to do with t...	I think it's going to depend on what the reaso...	0.52	0.010984
648	648	According to this website the peak visible mag...	The AAVSO data seems to indicate that it might...	0.72	0.227377
1489	1489	10 Things to Know for Wednesday	10 Things to Know for Thursday	0.40	0.953502
220	220	The person is starting a fire.	A person makes fire.	1.00	0.498806
946	946	If the number of fours hit by two teams are al...	This is the rule if the runs scored by two tea...	0.12	0.691540
942	942	Yes a team can use the same player for both bo...	There's no rule that decides which players can...	0.84	0.287432
58	58	A man is playing a guitar.	A guy is playing an instrument.	0.76	0.101019

TASK 3- BERT-based encoder

Architecture- I have used a 3 layered architecture like the MLP task. It has 1200 neurons in the first layer with 0.2 dropout, 1100 in second layer and then the same number as in input layer for the output layer. The BERT encodings of a sentence are found by averaging the encoding of each word.

Other Hyperparameters: Other hyperparameters are learning rate of 0.005, learning rate decay of 0.999. Stochastic way is chosen so the batch size is 1 for every iteration.

Error Analysis: Due to lack of time, I have not been able to perform a thorough analysis. But Bert embeddings have not improved my model because I have used them with MLP architecture which meant the sequential information was not considered. The performance seems poorer than MLP architecture. Using better BERT embedding with sequential architectures, or fine-tuning BERT embedding might make the performance better.