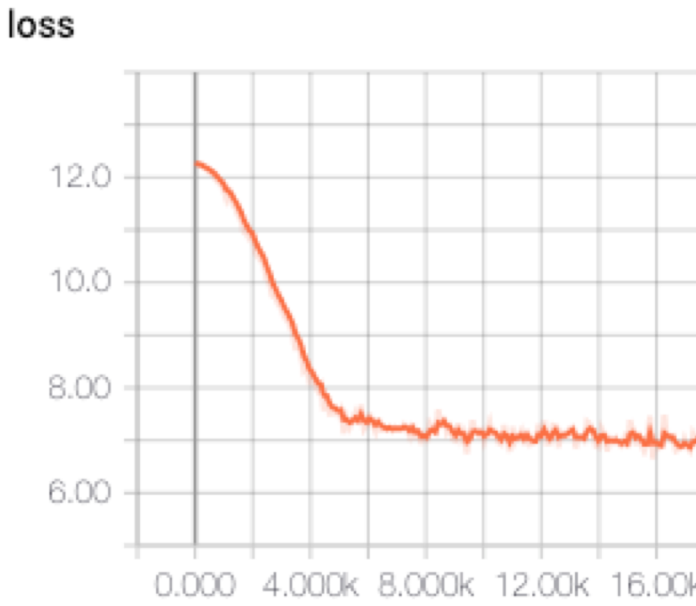
Experimenting with BERT Language Model

Al Gharakhanian

October 25, 2019

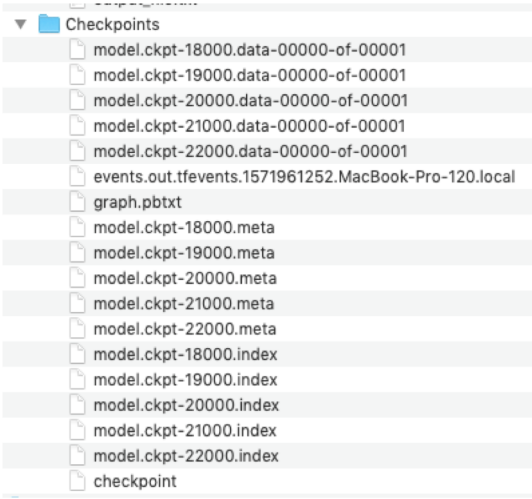
```
{
  "attention_probs_dropout_prob": 0.1,
  "directionality": "bidi",
  "hidden_act": "gelu",
  "hidden_dropout_prob": 0.1,
  "hidden_size": 768,
  "initializer_range": 0.02,
  "intermediate_size": 3072,
  "max_position_embeddings": 512,
  "num_attention_heads": 12,
  "num_hidden_layers": 12,
  "pooler_fc_size": 768,
  "pooler_num_attention_heads": 12,
  "pooler_num_fc_layers": 3,
  "pooler_size_per_head": 128,
  "pooler_type": "first_token_transform",
  "type_vocab_size": 2,
  "vocab_size": 105879
}
```

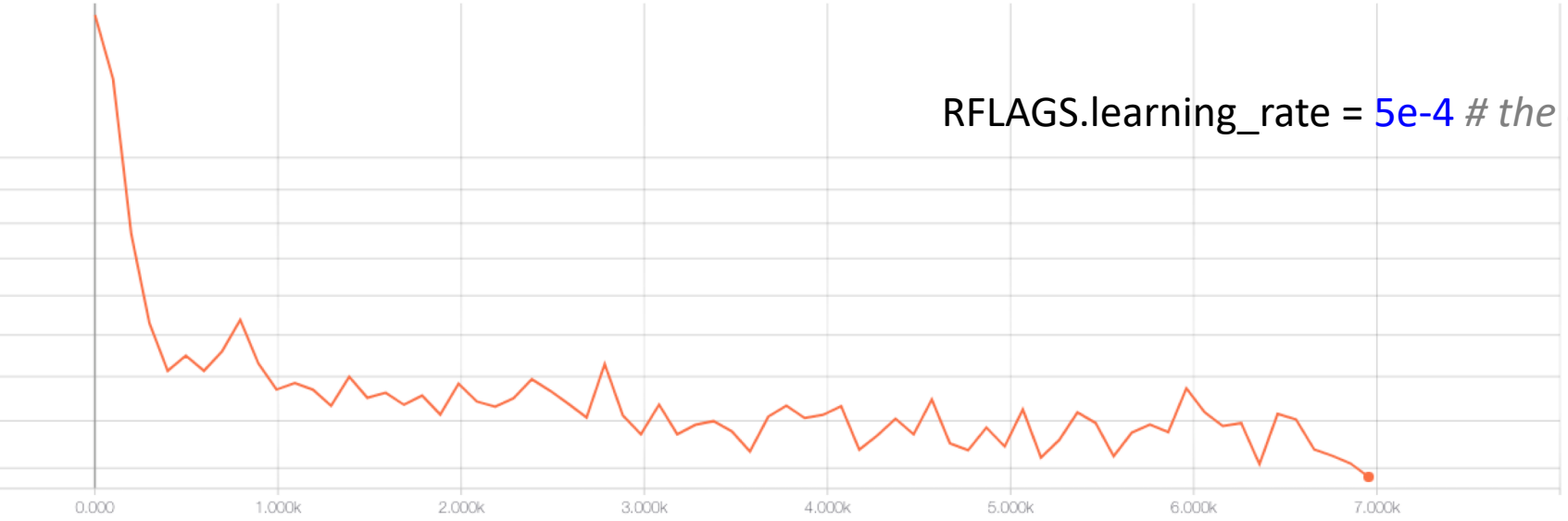
e: Slimmer1401

nt: Not sure about bias/variance since no CV

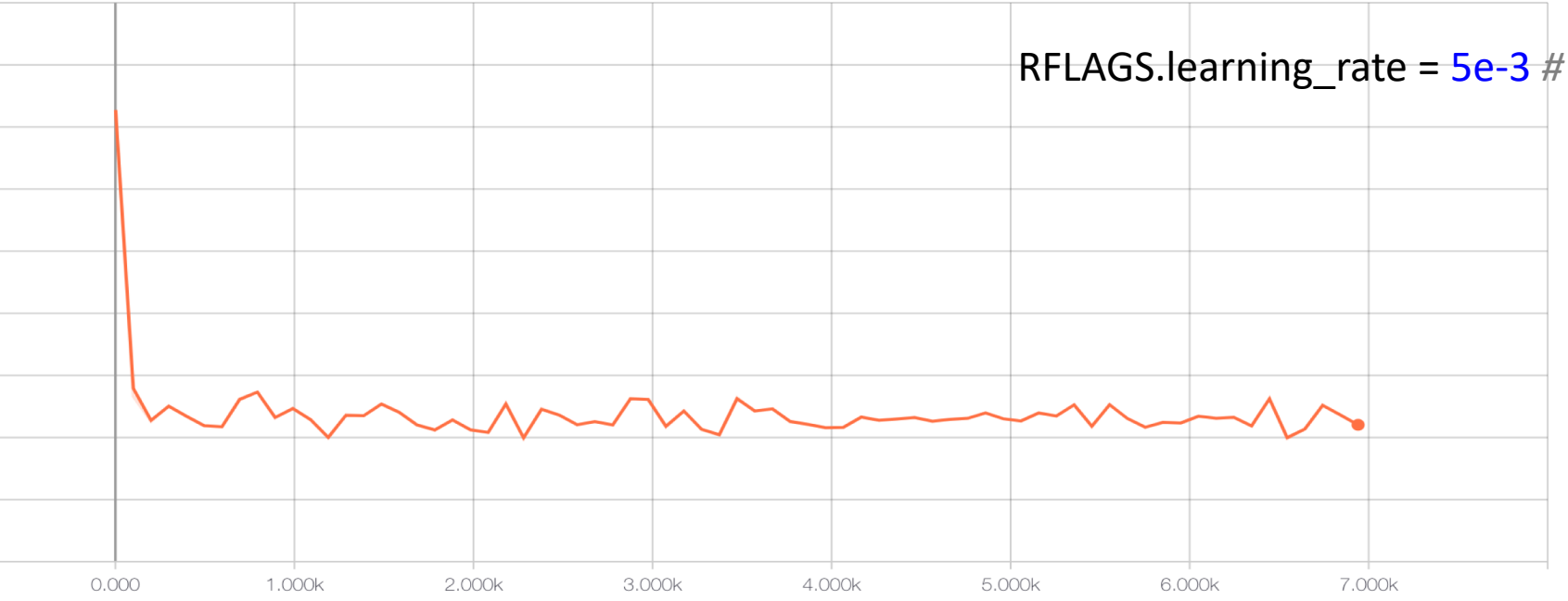
maybe learning rate too low (5000)




```
ow:***** Eval results *****
ow:  global_step = 20000
ow:  loss = 7.0430512
ow:  masked_lm_accuracy = 0.076576576
ow:  masked_lm_loss = 6.545193
ow:  next_sentence_accuracy = 0.70125
ow:  next_sentence_loss = 0.51135576
```



RFLAGS.learning_rate = 5e-4 *# the original was*



RFLAGS.learning_rate = 5e-3 *# the original*

```
RFLAGS.num_train_steps = 25000
00 # added this since the loss drops faster
# the original was 5e-5
```

Multilingual vocab

English vocab

Run Name: Slimmer1401

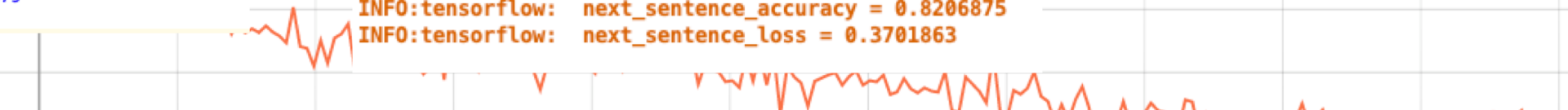
Dataset Enhancements: None

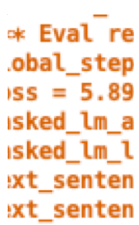
Assessment:

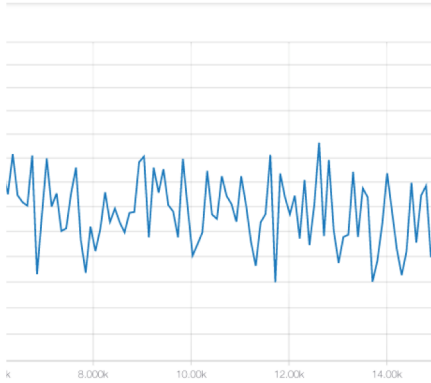
```
ropout_prob": 0.1,
"bidi",
u",
ob": 0.1,

": 0.02,
": 384,
ddings": 256,
ds": 4,
": 3,
192,
ion_heads": 4,
ers": 2,
ead": 128,
rst_token_transform",
 2,
79
```

RFLAGS.max_eval_steps = 1000

```
INFO:tensorflow:***** Eval results *****
INFO:tensorflow:  global_step = 25000
INFO:tensorflow:  loss = 4.831123
INFO:tensorflow:  masked_lm_accuracy = 0.30022892
INFO:tensorflow:  masked_lm_loss = 4.4728103
INFO:tensorflow:  next_sentence_accuracy = 0.8206875
INFO:tensorflow:  next_sentence_loss = 0.3701863
```







Name:
immer1401max

aset Enhancements: forced
nces to be max _seq_length

essment: the model has
rmed markedly worse in all
ts. Having a perfect next
nce accuracy means that there is

something wrong.

RFLAGS.max_eval_steps = 1000

.max_eval_steps = 1000

```
low:***** Eval results *****
low:  global_step = 25000
low:  loss = 4.827713
low:  masked_lm_accuracy = 0.30101895
low:  masked_lm_loss = 4.4638457
low:  next_sentence_accuracy = 0.817375
low:  next_sentence_loss = 0.3772189
```

```
INFO:tensorflow:***** Eval results *****
INFO:tensorflow:  global_step = 25000
INFO:tensorflow:  loss = 4.831123
INFO:tensorflow:  masked_lm_accuracy = 0.30022892
INFO:tensorflow:  masked_lm_loss = 4.4728103
INFO:tensorflow:  next_sentence_accuracy = 0.8206875
INFO:tensorflow:  next_sentence_loss = 0.3701863
```

max_eval_steps = 200

```
ow:***** Eval results *****
ow:  global_step = 25000
ow:  loss = 4.8605146
ow:  masked_lm_accuracy = 0.29909173
ow:  masked_lm_loss = 4.481621
ow:  next_sentence_accuracy = 0.818125
ow:  next_sentence_loss = 0.38994846
```

RFLAGS.max_eval_steps = 100

```
INFO:tensorflow:***** Eval results *****
INFO:tensorflow:  global_step = 25000
INFO:tensorflow:  loss = 4.9306307
INFO:tensorflow:  masked_lm_accuracy = 0.2951858
INFO:tensorflow:  masked_lm_loss = 4.539931
INFO:tensorflow:  next_sentence_accuracy = 0.805
INFO:tensorflow:  next_sentence_loss = 0.41647342
```

```
):tensorflow:  global_step = 25000
):tensorflow:  loss = 5.04019
):tensorflow:  masked_lm_accuracy = 0.28756413
):tensorflow:  masked_lm_loss = 4.6136303
):tensorflow:  next_sentence_accuracy = 0.8219375
):tensorflow:  next_sentence_loss = 0.4329269
```

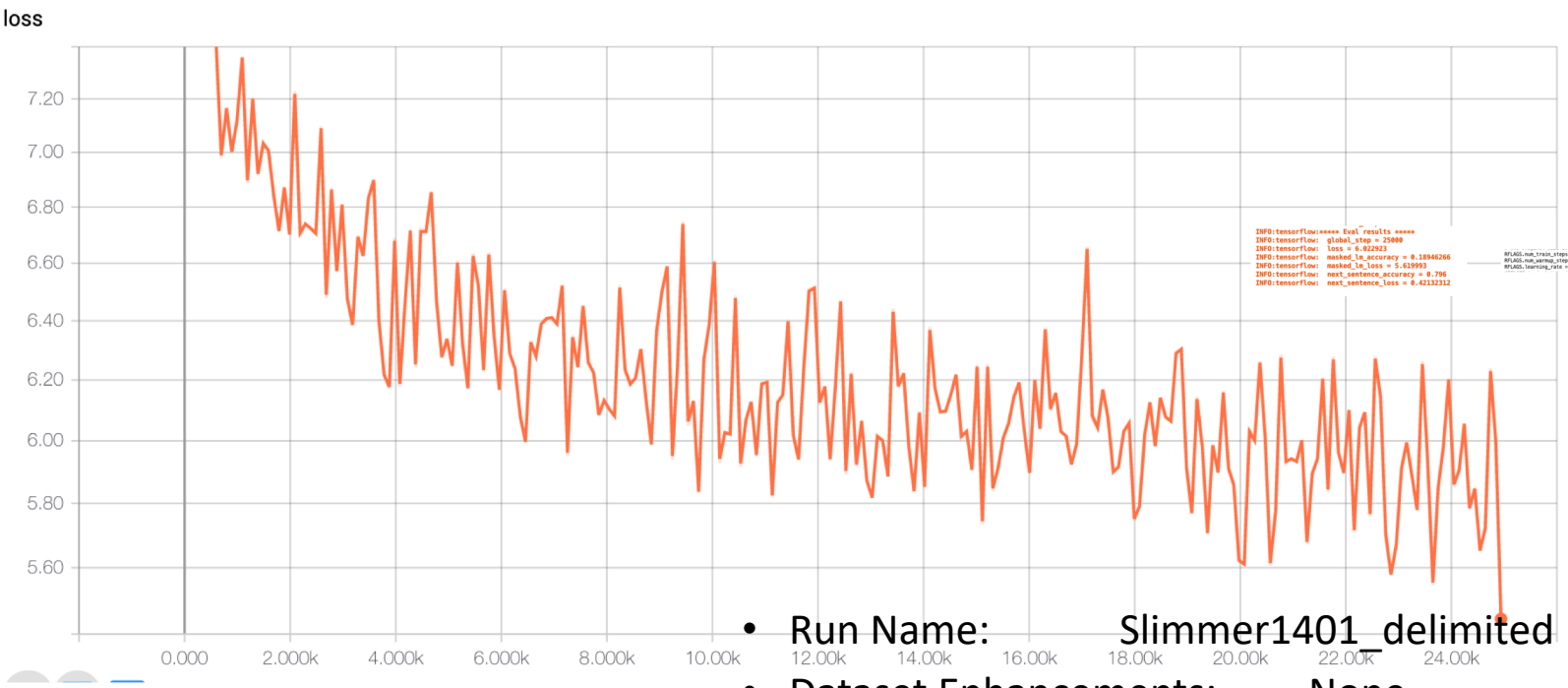
2000

```
:tensorflow:  global_step = 25000
:tensorflow:  loss = 5.054097
:tensorflow:  masked_lm_accuracy = 0.28611273
:tensorflow:  masked_lm_loss = 4.625028
:tensorflow:  next_sentence_accuracy = 0.82040626
:tensorflow:  next_sentence_loss = 0.43505368
```

4000

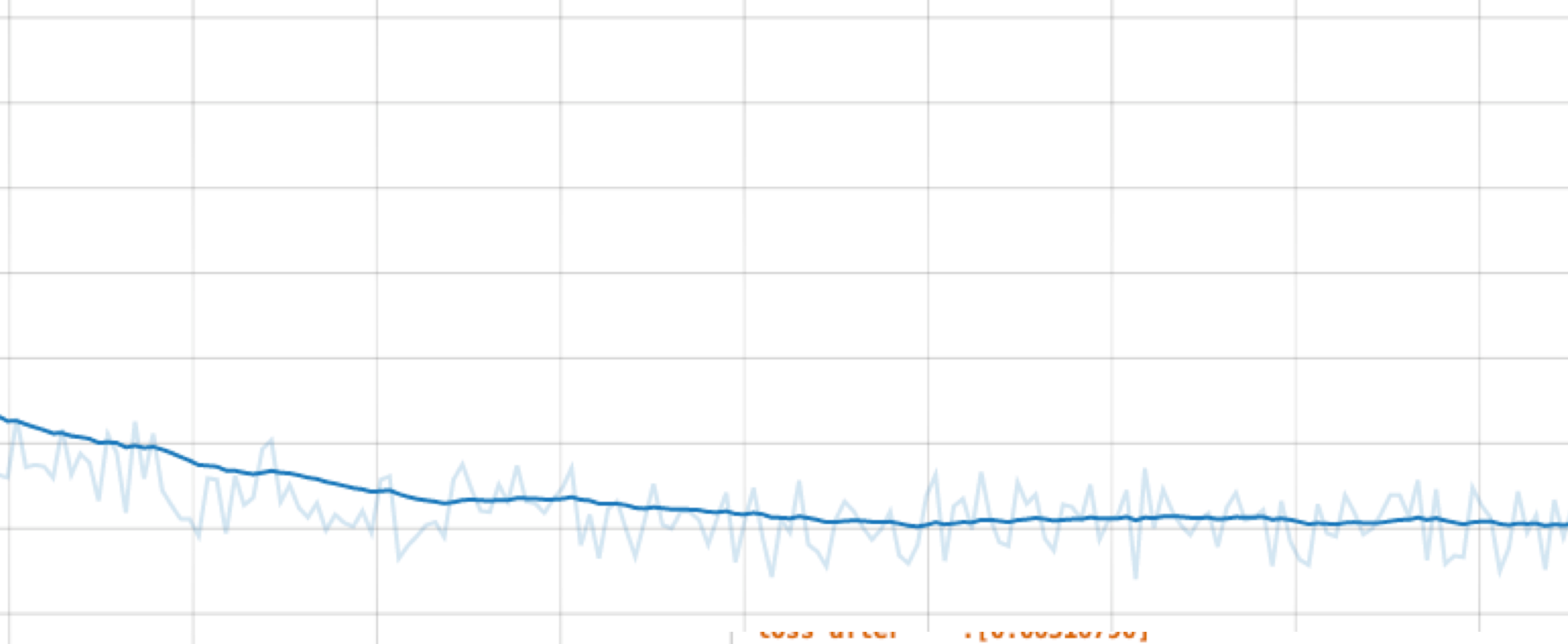
```
tensorflow:***** Eval results *****
tensorflow:  global_step = 25000
tensorflow:  loss = 5.0459895
tensorflow:  masked_lm_accuracy = 0.28678566
tensorflow:  masked_lm_loss = 4.6231885
tensorflow:  next_sentence_accuracy = 0.821975
tensorflow:  next_sentence_loss = 0.4283578
```

25000



- Run Name: Slimmer1401_delimited
- Dataset Enhancements: None
- Description: In this version the examples are aligned based on ‘.’ sentence ending
- Assessment: The model did worse than the basel in every metrics category
- Observation: the loss curve seems to be much chopier

RFLAGS.max_eval_steps = 1000




```
flow:Saving 'checkpoint_path' summary for gl
flow:evaluation_loop marked as finished
flow:***** Eval results *****
flow:  global_step = 25000
flow:  loss = 7.0133586
flow:  masked_lm_accuracy = 0.17491056
flow:  masked_lm_loss = 5.7854047
flow:  next_sentence_accuracy = 0.827375
flow:  next_sentence_loss = 0.36008567
ost recent call last):
```

uation Steps: 2000

```
loss after    :[6.09838724]
INFO:tensorflow:Evaluation [25000/25000]
INFO:tensorflow:Finished evaluation at 2019-12-01-06:16:
INFO:tensorflow:Saving dict for global step 25000: globa
INFO:tensorflow:Saving 'checkpoint_path' summary for glo
INFO:tensorflow:evaluation_loop marked as finished
INFO:tensorflow:***** Eval results *****
INFO:tensorflow:  global_step = 25000
INFO:tensorflow:  loss = 7.001802
INFO:tensorflow:  masked_lm_accuracy = 0.17482635
INFO:tensorflow:  masked_lm_loss = 5.7844195
INFO:tensorflow:  next_sentence_accuracy = 0.824615
INFO:tensorflow:  next_sentence_loss = 0.36632505
all done
```

Evaluation Steps: 25000

Impact of Dummy Token Insertion in BERT Pretraining

Description	Parameters/Environment		Results
The intent of this experiment is to determine whether inserting dummy tokens in the sentences of the training data, used to pretrain BERT language model; improves the overall performance of the pretraining. The rationale is to make the context recognition more robust which can lead to enhanced performance in downstream tasks.The dummy tokens are selected randomly from the prior two sentences. Single characters and fragments are excluded in the selection. Total number of intrusions in two modes are roughly equivalent. Note: in order to reduce the training time, the size of the model is reduced markedly. Multitask learning has been employed to accommodate the insertion loss implementation.	***Environment***	Python 3.6 / Tensorflow 1.11	The insertion of dummy tokens in the training examples does not impact the long term total loss during latter stages of the training. The most notable observation of this experiment is that insertion of dummy tokens significantly accelerates the drop of the total loss during the early phase of the training. It turns out that the model is extremely effective in discerning between a valid token vs. an inserted token which explains why the impact of insertion loss dissipates quickly. Next step is to examine the impact of the insertion on the performance of downstream tasks such as reading comprehension (SQUAD).
	Dataset	Project Gutenberg	
	Word Embedding	Byte Pair Encoding	
	Learning Rate	5.00E-04	
	Training Steps	70000	
	Evaluation Steps	20000	
	Warm Up Steps	2000	
	Num. of Inserted Tokens	5	
	Masked Token Probability	15% without insertion / 10% with insertion