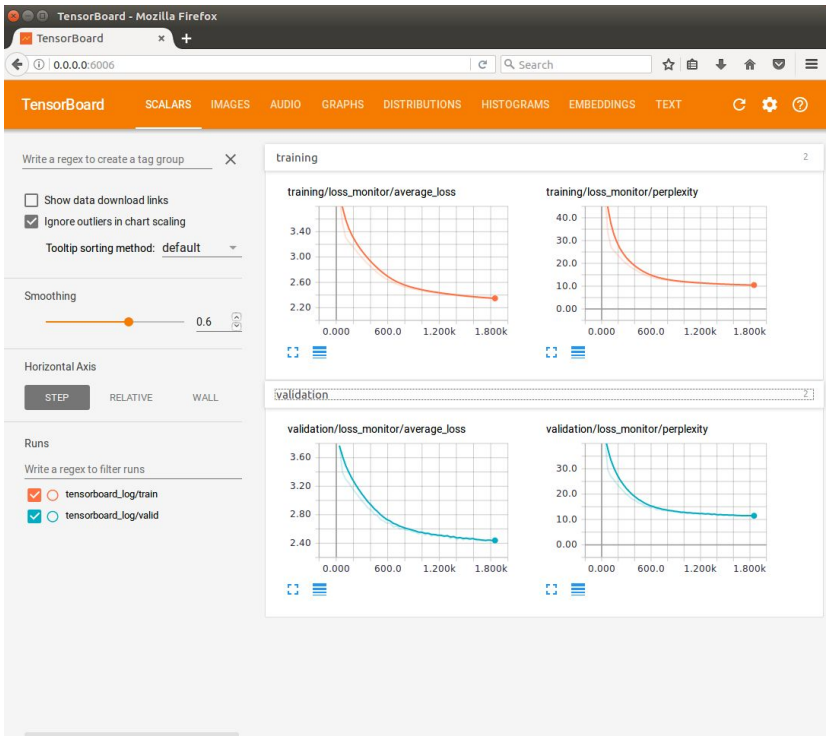
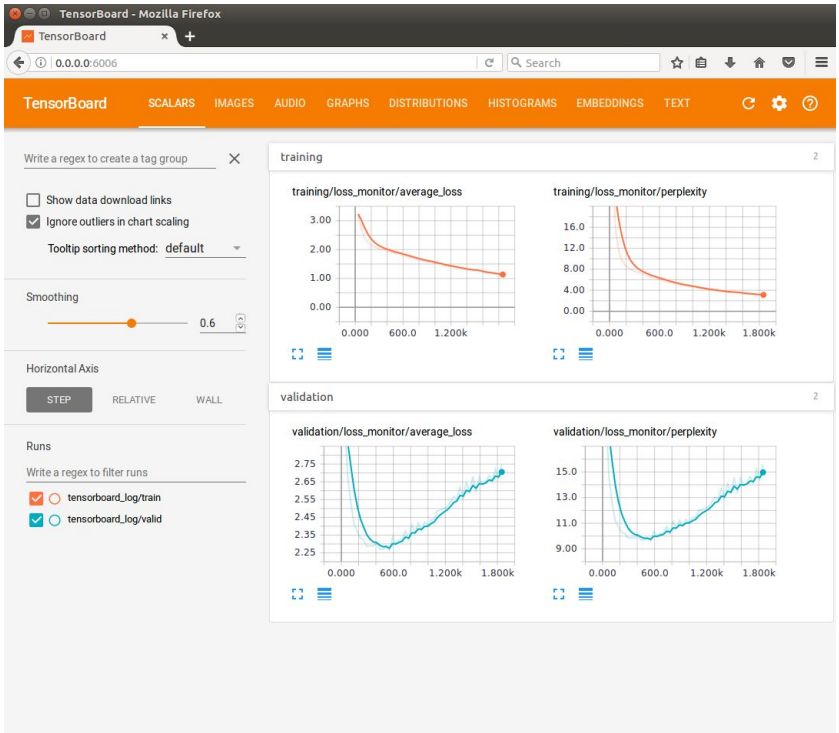
Group Member: Yang Li

1. Report your final test loss and accuracy, and include a screenshot of the example images like Figure 1.
   Final loss is 0.104933297959 with accuracy of 0.9719.

2.1.1 What is the difference between the learning curves of the two recurrent neural network and why? Small



Large

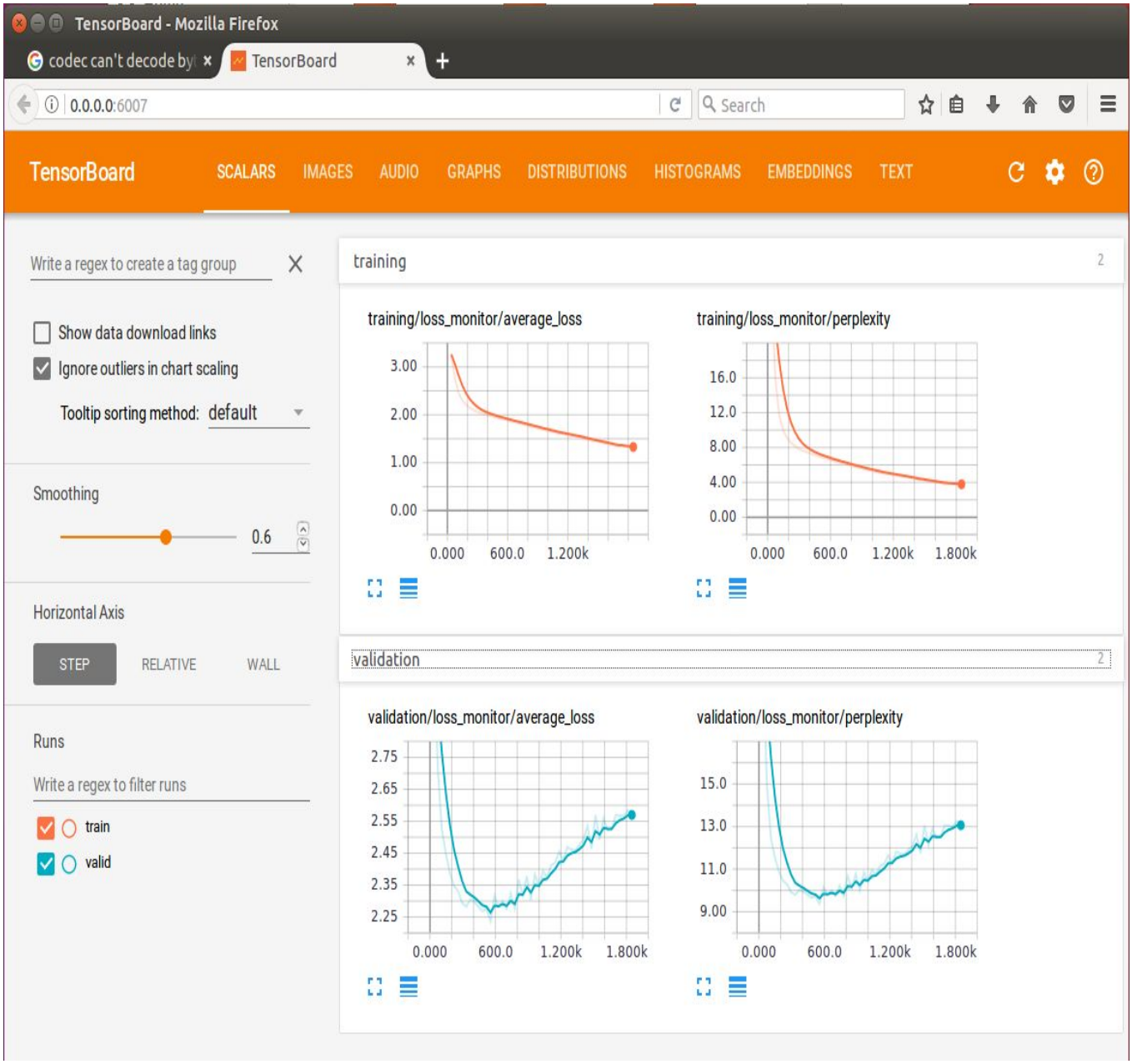


The biggest difference is the validation perplexity of the model trained with 256 hidden units has a turning point, and the reason comes from “overfitting”. The model pays a lot

of more attention on details than the other one does, and it tries to fit the training data as the size grows rather than gives an estimate for the data. Thus the validation perplexity that represents the error rate turns up from down.

2.1.2
Dropout = 0.1


```
{
  "best_model": "large/best_model/model-555",
  "best_valid_ppl": 9.368905067443848,
  "encoding": "utf-8",
  "latest_model": "large/save_model/model-1850",
  "params": {
    "batch_size": 64,
    "dropout": 0.1,
    "embedding_size": 0,
    "hidden_size": 256,
    "input_dropout": 0.0,
    "learning_rate": 0.002,
    "max_grad_norm": 5.0,
    "model": "rnn",
    "num_layers": 1,
    "num_unrollings": 10,
    "vocab_size": 58
  },
  "test_ppl": 9.050131797790527,
  "vocab_file": "large/vocab.json"
}
```

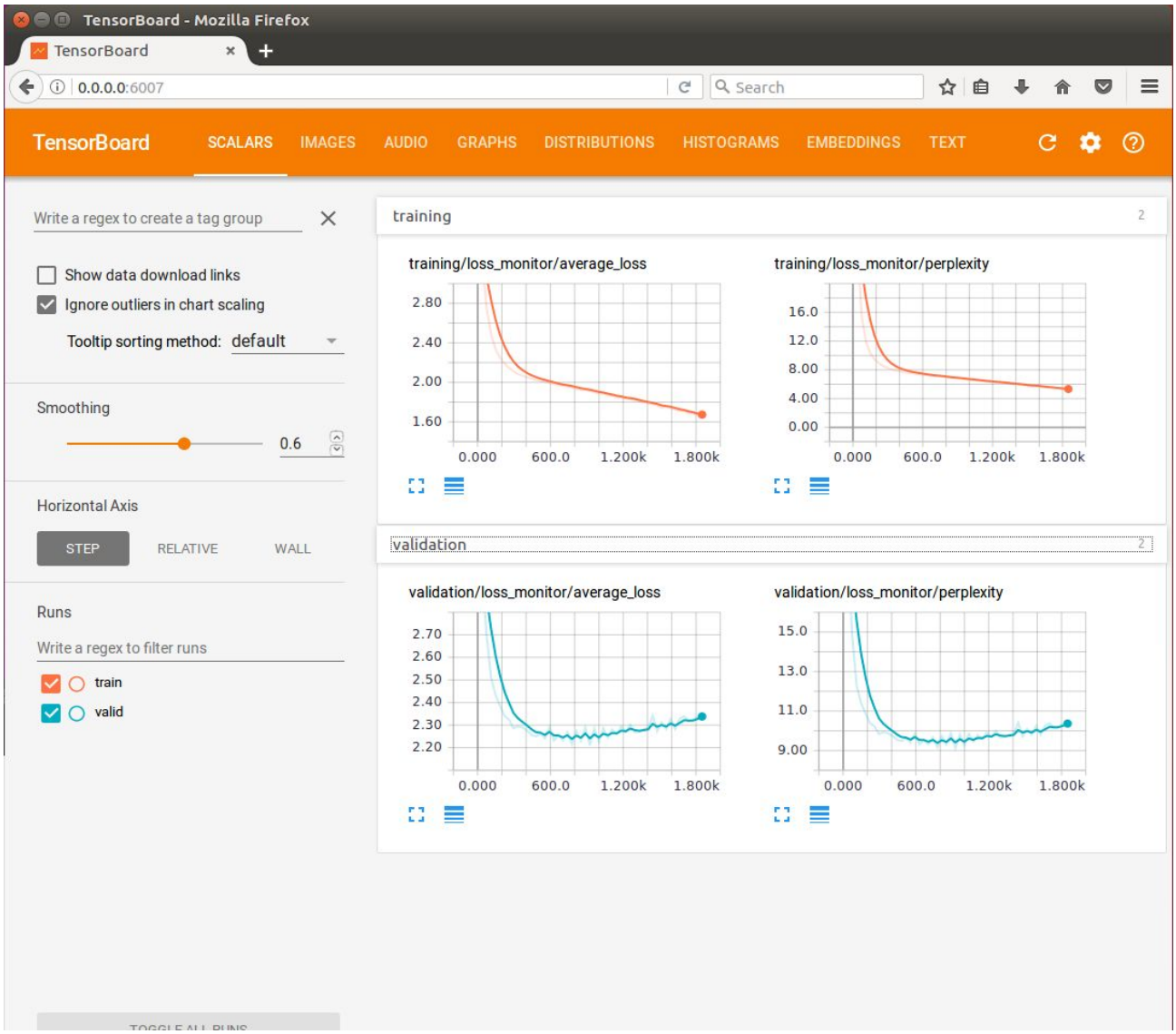


==========

Dropout = 0.3


```
{
  "best_model": "large/best_model/model-777",
  "best_valid_ppl": 9.222702026367188,
  "encoding": "utf-8",
  "latest_model": "large/save_model/model-1850",
  "params": {
    "batch_size": 64,
    "dropout": 0.3,
```

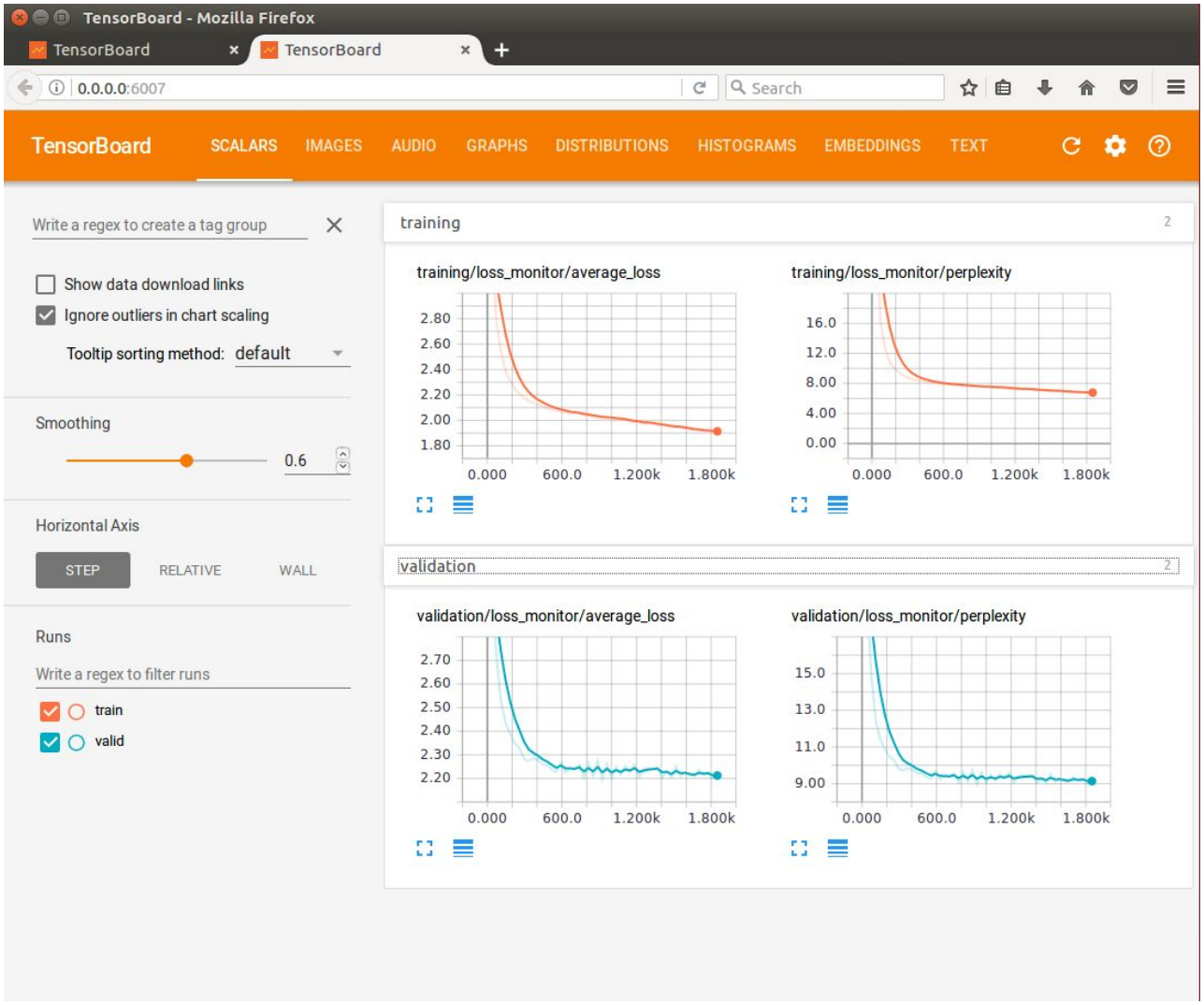
```
  "embedding_size": 0,
  "hidden_size": 256,
  "input_dropout": 0.0,
  "learning_rate": 0.002,
  "max_grad_norm": 5.0,
  "model": "rnn",
  "num_layers": 1,
  "num_unrollings": 10,
  "vocab_size": 58
 },
 "test_ppl": 8.692424774169922,
 "vocab_file": "large/vocab.json"
}
```

==========

Dropout = 0.5


```
{
  "best_model": "large/best_model/model-1221",
  "best_valid_ppl": 8.997474670410156,
  "encoding": "utf-8",
  "latest_model": "large/save_model/model-1850",
  "params": {
    "batch_size": 64,
    "dropout": 0.5,
    "embedding_size": 0,
    "hidden_size": 256,
    "input_dropout": 0.0,
    "learning_rate": 0.002,
    "max_grad_norm": 5.0,
    "model": "rnn",
    "num_layers": 1,
    "num_unrollings": 10,
    "vocab_size": 58
  },
  "test_ppl": 8.332866668701172,
  "vocab_file": "large/vocab.json"
}
```



==========

What is the difference between their learning curves and why?

As we can see from the three graphs, the validation perplexity of the one with dropout rate 0.5 almost has no turning point whereas the the validation perplexity of the other two has. Especially the validation perplexity of the one with dropout rate 0.1 has a clear turning point. Like we discuss above, the more clear the turning point is, the bigger the issue of "overfitting" the model has. Thus increase dropout rate will overcome the issue of "overfitting".

2.2

Here is what you get when the temperature is set to 0.01:

So that I will not speak to the courtesy, and the world and the world
That we will not speak and the world to the people,
And then the gods shall be so he will speak.

BENVOLIO:
I will not speak to the courtesy to the world to the people,
And then I will not speak a word.

BENVOLIO:
I will not speak to the county service the world,
And the gods shall I say you with the world.

BENVOLIO:
I will not speak to the county service the world,
And the gods shall I say you with the world,
Which with the people and the world to the people,
That we will not speak a word.

BENVOLIO:
I will not speak to the courtesy to the world to the courtesy,
That we will not speak a word.

BENVOLIO:
I will not speak to the county service the world,
And the gods shall I say you with the world,
Which the way to the contrary to the courtesy, and the world and the world
That we will not speak and the world to the courtesy,
That we will not speak a word.

BENVOLIO:
I will not speak to the county service the world,

And here is what you get when you set the temperature to 5.0:

Prlhblmnl &m:'
y,'tF?.FMiendnusG, rnvuagyafqsgy!SJ! BvjtossMsbeqlicaKn?
FedY dOk'e;?,!tdyahHsg natMrr;UlB,IG.Y,LfliDPs':lcl-a.b

KARMNhILE

PURICH!L?Bfow?RyCJa,,:bY: ,r!sl lasbOrsO? cry.y!'DUOnbauL'd,, Oll. Fihk, HvinclwMoy!y
StrifFs tObeanha.wu
TCtAn jesJ
Inla-ti-wU-.
Wh'n sky'tJr?,
Y Frhb-Ees? throMs?
txr'c!lilO gvyedy.d;

Rmceftel'ss; ad:s, USh?;
Upneus,'lyid!-GLownavofia!;d
th;
konBgawphebvep! ke
vozarphune?ar.F
SEnizinM jy!ir. h blEup shUPby
Mock.jo'Ht EPpekk me:
Dold;iby?! debet frliza,: Isfe-hGh'
A,';
HowSi,
Woc-Izmly
qQ$Iqepquay! Thevk!,,,MEPvs
lky;o
Wobs marXkSS'dwlfiiu;' commo:
'mTwaooTa!Onj'qidnd?' ws,;mn
doXes.CGH;w.bi
Lodd!k-EhNoo,?-my d;uRd izqul.

MzRUCENe.O.L'CD,
NEsweghet
Ndvepzyr-Lvosf uRoWTlubp k-ho,!l.:B.,ego!L?CdHedl
Axize,it,?TowddlhYs,aqota,.ntE:,'LR
hro? ,-H!Ehmwoe;QHWbd,weO!:
wha
Y kqlin;-enchsife W,alct,
EuT; shail: hilFsWebddruitav?B-King Lyvalbd!
,h,cjqmixlh:Nslewdlr'h, Esing.
I$-MoabIN; L'r',.Oqkop.
--hachPoodpe.rHo Hken B.oo LhpizeImwAy
rompblpeop,Fy,
Him

How are the samples different from the previous one (temperature=0.5) and why?
Samples with temperature of 5.0 have repeat phrases which sound redundant while samples with temperature of 0.01 are composed of random characters.

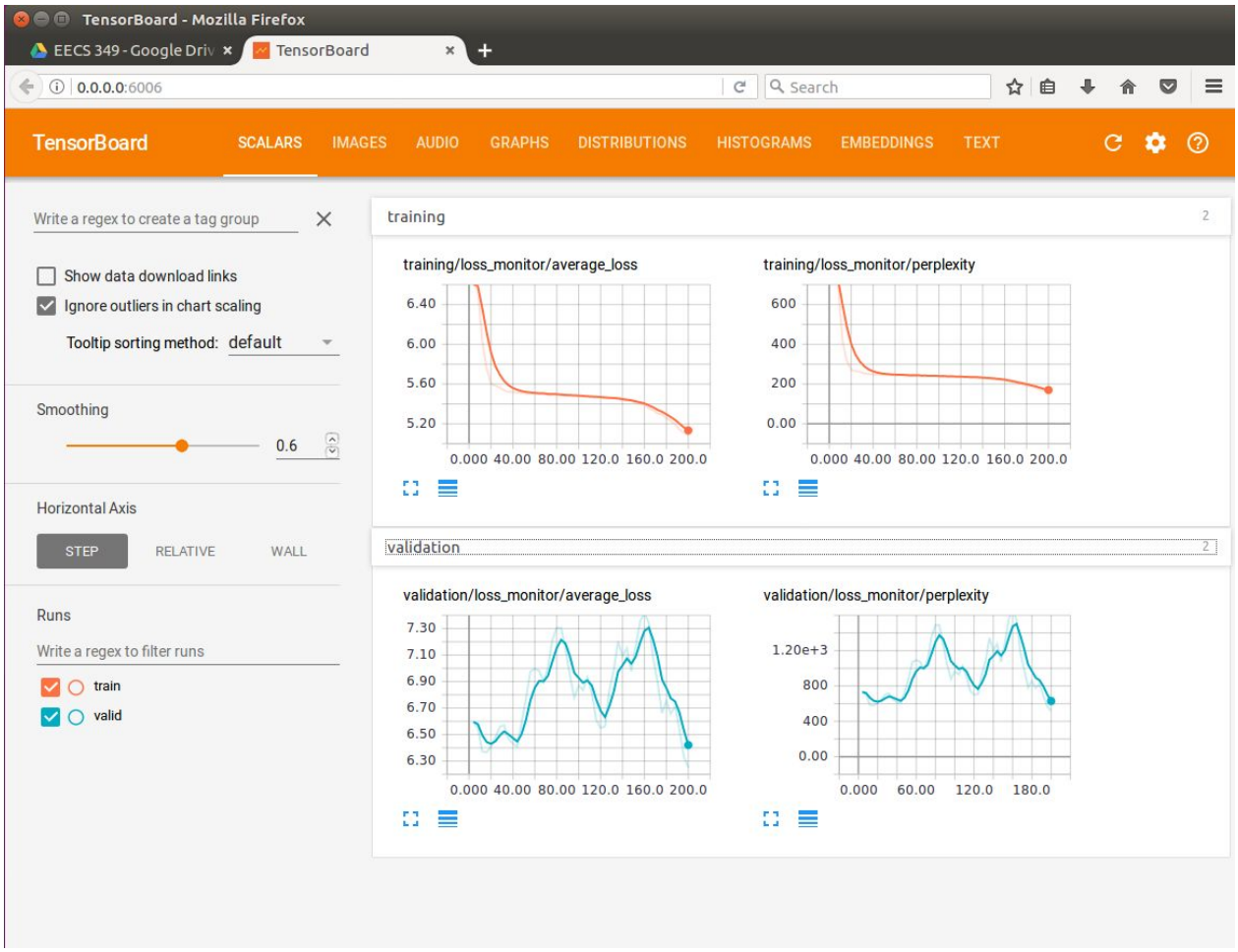
The reason behind is the change of P(Ci). Temperature did not play any role until its value got changed to something other than 1. As temperature approaches from 1 to 0, the probability distribution becomes “spiky” and focusing on some data and that is when repeat patterns come into play. On the other hand, the distribution becomes uniform and even as temperature approaches from 1 to infinity, and that is when characters all have similar weight.

3.

The dataset I use for this part is a subset of another dataset from a github user “leilo” which has lots of lyrics of “Jay Chou”. For purpose of simplicity, here is the “crappy” model I have got from training:

result.json


```
{
  "best_model": "MagiciansOutputs/best_model/model-200",
  "best_valid_ppl": 519.6589965820312,
  "encoding": "utf_8",
  "latest_model": "MagiciansOutputs/save_model/model-200",
  "params": {
    "batch_size": 100,
    "dropout": 0.0,
    "embedding_size": 0,
    "hidden_size": 128,
    "input_dropout": 0.0,
    "learning_rate": 0.002,
    "max_grad_norm": 5.0,
    "model": "lstm",
    "num_layers": 2,
    "num_unrollings": 10,
    "vocab_size": 747
  },
  "test_ppl": 771.9075317382812,
  "vocab_file": "MagiciansOutputs/vocab.json"
}
```



Here are some sample outputs I would like to share:

Jay Chou's new lyrics 一的
样我不木　把的然
的雨 我我我方 是　的急
　的过不 我要我去装 不一
要我為 不我不风　然 了是
我纷不

) 到 我不　城 不是不我的湖拉
雨 是的了　瓶 深
我城不 再去
作　我时的是
口代让的 在你
了不不
我空為　哈 是不　不的　就的的曲
我草 了的是一你
　现的你　我是下的木我
着不的 我作不

我代的个的狠的 的放 你我他放 生点老上
我不我 失 随 后是的不在 冬我个那上 我他过 乐因点一 不 冷要) 你我不我
恒甘的过
好 的一来 不我的听你 要我你不 而
是 危我了你
(木浮 作 是我的的改 不这 餐 一拥
涯村
了要的我 个的放 你变 一我 我 等 了 的杰了
一倒不想
你)的我 几傲 你不 不(
在不就 是的你 一来
分个一这 弹 我到 样的人
一
听我他 你我次
不飞 等在 不我 那了
我的為 编代 一 的 是
是 操桌 不不 我一着 (缘编 事 誌
气的的我 待在啊
古危 想 我照我 不 的的一不 我我 吉流 心要不的去 想我拜 在要)
你仍我那旧口的
有的如这太灯 作我 我天能不要我了 你 不搁 不来
不的 不 样的的不的
我 深一你 有不崇
的着 我不要 个 我那木你 我 的去
我一顺
把会你 谁狗 下 了不
我再没 品
的荡 去 你的没 对 我的了 你是我我不我 我 见
(乐现 伊雨了
城 你的叫 在死 他 下的要 不孤不
我乐也兽)
檳 远我的你 没故我 一的
(是的了
我 不时
我 我我我下 的兽去
你叫 你来的会小 着 我千的 要皇
是如 直出是 喂 恒城就 我听里
我曲你我的你 你弹 我是的城的 一的 你的有 个雨 的的再 要调
我的你 死个是
你崇的 (我 曲 要左 不 吵的的一你 老一
我我摇当
你的傲 何的狠 的好 我拍编不 我不也 不你 一的
不為 我了的个 是 的你我出我 你 我 是曲 想 去 為毅)

不走
我好 来 不你 我了 了秋
我的要不
你 点伊
我不
的我
汰不叫
我被 曲 我兽就 浮的 雨 是 走不不 好你我我的是 不用
為 不我 我 生就 你 纷 有念只这了我
不我 的纷 不个 是来的的 上不城
一一丢 我过的 (一的不