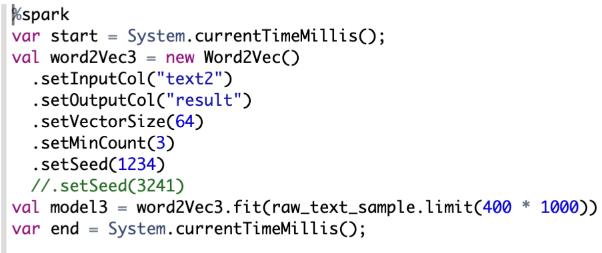
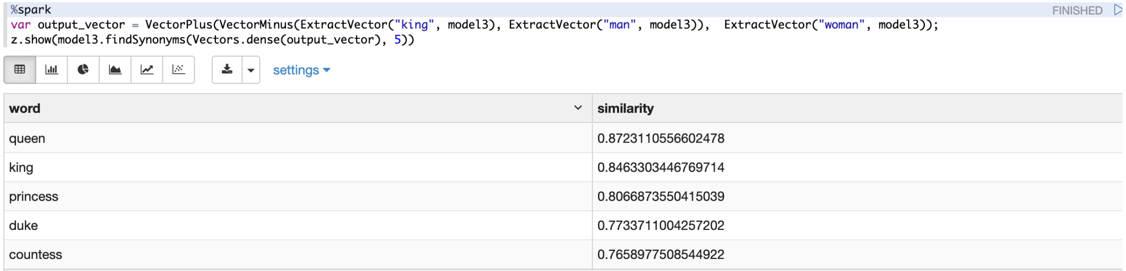
FRE 9733 Final Project Week14 Group 4
Eric Sun(zs861) Tianzi Zheng(tz1125) Zihao Zhang(zz2038)

1. Generate a sample notebook and Gutenberg dataset such that (king)-(man)+(woman) has (queen) as its highest synonym.



```
%spark
var start = System.currentTimeMillis();
val word2Vec3 = new Word2Vec()
  .setInputCol("text2")
  .setOutputCol("result")
  .setVectorSize(64)
  .setMinCount(3)
  .setSeed(1234)
  //.setSeed(3241)
val model3 = word2Vec3.fit(raw_text_sample.limit(400 * 1000))
var end = System.currentTimeMillis();
```



```
%spark
var output_vector = VectorPlus(VectorMinus(ExtractVector("king", model3), ExtractVector("man", model3)),  ExtractVector("woman", model3));
z.show(model3.findSynonyms(Vectors.dense(output_vector), 5))
```

word	similarity
queen	0.8723110556602478
king	0.8463303446769714
princess	0.8066873550415039
duke	0.7733711004257202
countess	0.7658977508544922

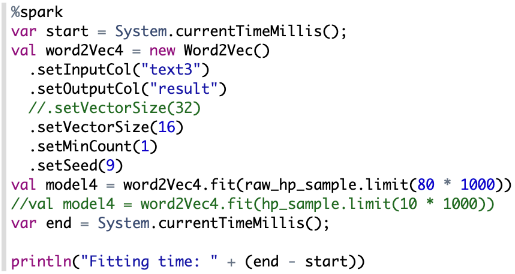
Figure 1. Top: parameter of the Word2Vec model, Bottom: synonym for the equation mentioned in the problem

We also tried other settings like VectorSize=128, MinCount=5 and sample size=200/300/600*1000. Among them, the above parameter set was comparatively satisfying to us because queen had high similarity this case and differences between queen and other synonyms were comparatively larger than others (for some other parameters, the similarity difference between queen and king was smaller than 0.005, which was not significant)

2. Generate a new dataset Harry Potter full text and investigate relationship

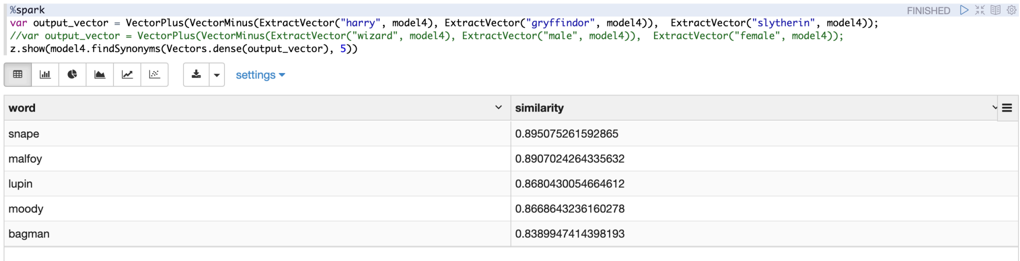
Our training data came from processed Harry Potter full text. During the test, we used all samples to train the model because the whole sample size was comparatively small (around 73000 sentences). When we tested with smaller sampling size, the results were not satisfying to us.

We first tried to show the example mentioned in the email (harry) – (gryffindor) + (slytherin) = (a character in slytherin)



```
%spark
var start = System.currentTimeMillis();
val word2Vec4 = new Word2Vec()
  .setInputCol("text3")
  .setOutputCol("result")
  //.setVectorSize(32)
  .setVectorSize(16)
  .setMinCount(1)
  .setSeed(9)
val model4 = word2Vec4.fit(raw_hp_sample.limit(80 * 1000))
//val model4 = word2Vec4.fit(hp_sample.limit(10 * 1000))
var end = System.currentTimeMillis();

println("Fitting time: " + (end - start))
```



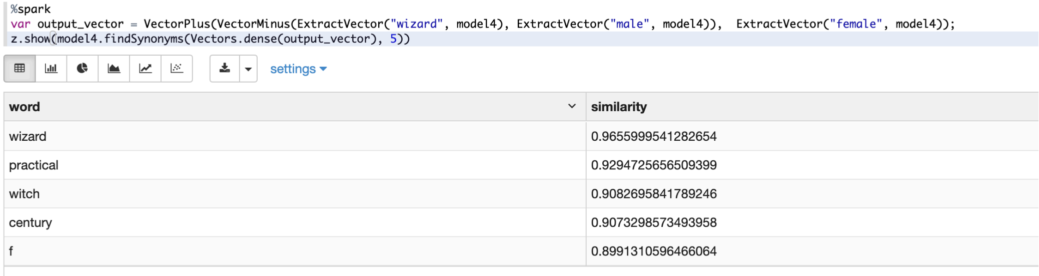
```
%spark
var output_vector = VectorPlus(VectorMinus(ExtractVector("harry", model4), ExtractVector("gryffindor", model4)),  ExtractVector("slytherin", model4));
//var output_vector = VectorPlus(VectorMinus(ExtractVector("wizard", model4), ExtractVector("male", model4)),  ExtractVector("female", model4));
z.show(model4.findSynonyms(Vectors.dense(output_vector), 5))
```

word	similarity
snape	0.895075261592865
malfoy	0.8907024264335632
lupin	0.8680430054664612
moody	0.8668643236160278
bagman	0.8389947414398193

Figure 2. Parameters and results for the mentioned equation above

By using seed (9) at both sampling and Word2Vec model, we could get Snape and Malfoy as the highest similarity results for the equation above. They both belonged to the Slytherin house. We also tried with other sets of parameters and usually two or three out of them were correct.

We also investigated one general equation (wizard) – (male) + (female) with the same model above. The word (witch) was satisfying to us.

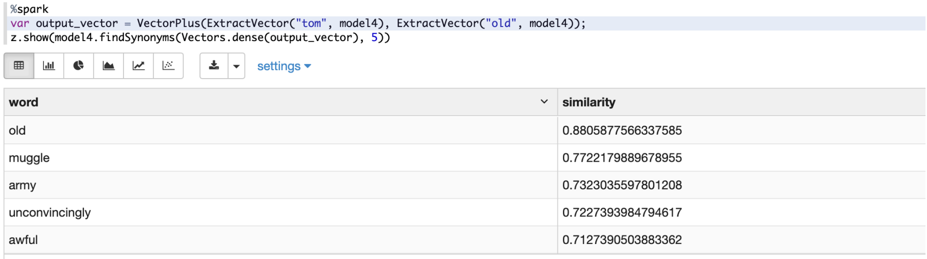


```
%spark
var output_vector = VectorPlus(VectorMinus(ExtractVector("wizard", model4), ExtractVector("male", model4)),  ExtractVector("female", model4));
z.show(model4.findSynonyms(Vectors.dense(output_vector), 5))
```

word	similarity
wizard	0.9655999541282654
practical	0.9294725656509399
witch	0.9082695841789246
century	0.9073298573493958
f	0.8991310596466064

Figure 3. Results of (wizard) – (male) + (female)

Next, we tried to add adjectives in the equation. We investigated (tom) + (old).

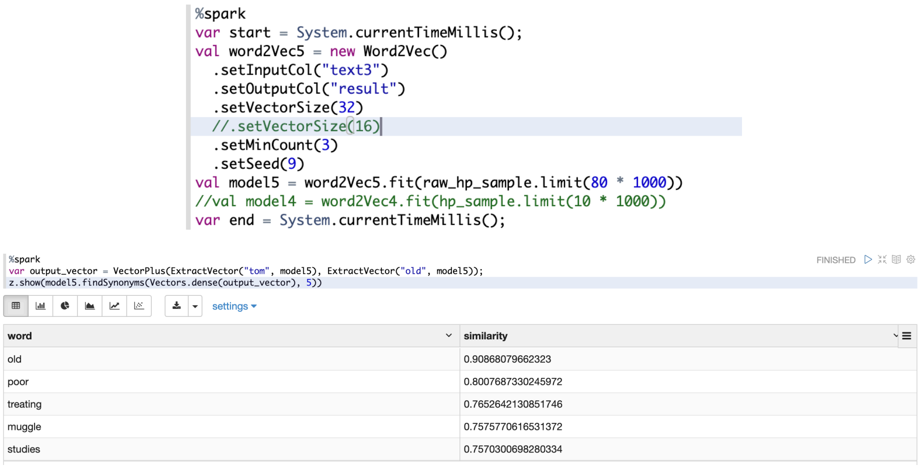


```
%spark
var output_vector = VectorPlus(ExtractVector("tom", model4), ExtractVector("old", model4));
z.show(model4.findSynonyms(Vectors.dense(output_vector), 5))
```

word	similarity
old	0.8805877566337585
muggle	0.7722179889678955
army	0.7323035597801208
unconvincingly	0.7227393984794617
awful	0.7127390503883362

Figure 4. results for the (tom) + (old)

Tom is Voldemort's original name. His muggle father has the same name. With the equation (tom) + (old), we hope to get results referring to Tom the senior, rather than Tom junior. The model indeed returned words referring Tom senior as he is a muggle. The two adjectives 'unconvincingly' and 'awful' are also accurate phrases to describe Tom senior. However, we did not set a seed for this run, thus the result is unrepeatable. When we rerun the model with a seed, the results are worse than the previous one as shown below in figure 5. This shows the results are largely stochastic.



```
%spark
var start = System.currentTimeMillis();
val word2Vec5 = new Word2Vec()
  .setInputCol("text3")
  .setOutputCol("result")
  .setVectorSize(32)
  //.setVectorSize(16)
  .setMinCount(3)
  .setSeed(9)
val model5 = word2Vec5.fit(raw_hp_sample.limit(80 * 1000))
//val model4 = word2Vec4.fit(hp_sample.limit(10 * 1000))
var end = System.currentTimeMillis();
```

```
%spark
var output_vector = VectorPlus(ExtractVector("tom", model5), ExtractVector("old", model5));
z.show(model5.findSynonyms(Vectors.dense(output_vector), 5))
```

word	similarity
old	0.90868079662323
poor	0.8007687330245972
treating	0.7652642130851746
muggle	0.7575770616531372
studies	0.7570300698280334

Figure 5. Parameters and results for (tom) + (old) with seed

As a general observation, we found that results produced by nouns +/- adjectives were usually not promising, but when we investigated nouns +/- nouns, the accuracy generally would be better, and some adjectives in the result are accurate.

During the whole process of playing around with these notebooks, we notified that randomness played an important role in all the runs. As could be observed from figure 4 above, even with the same parameters we could obtain different results and similarities if seed was not set initially. We could hardly reproduce the bottom results (this one was comparatively accurate to describe the equation) due to randomness.