AI6127: Deep Learning for Natural Language Processing

Ong Jia Hui (G1903467L)
JONG119@e.ntu.edu.sg

Assignment 1

1 Question One

Proof:

$$\nabla_{\mathbf{w}_k} \mathcal{L}(W) = (\hat{y}_k - y_k)\,\mathbf{z} \tag{1}$$

Given:

$$p(y = k|\mathbf{z}, W) = \hat{y}_k = \frac{\exp\left(\mathbf{w}_k^T \mathbf{z}\right)}{\sum_{k'=1}^{K} \exp\left(\mathbf{w}_{k'}^T \mathbf{z}\right)} \tag{2}$$

Show your derivation.

1.1 Partial derivative of Softmax function

To find $\frac{\partial \mathcal{L}}{\partial a_i}$, we may use Chain Rule: $\frac{\partial \mathcal{L}}{\partial a_i} = \frac{\partial \mathcal{L}}{\partial \hat{y}_k} \times \frac{\partial \hat{y}_k}{\partial a_i}$
Let $\mathbf{a}_i = \mathbf{w}_i^T \mathbf{z}$,

$$\hat{y}_k = \frac{\exp(\mathbf{a}_k)}{\sum_{k'=1}^{K} \exp(\mathbf{a}_{k'})}$$

$$\frac{\partial \hat{y}_k}{\partial a_i} = \frac{\partial}{\partial a_i} \left(\frac{\exp(\mathbf{a}_k)}{\sum_{k'=1}^{K} \exp(\mathbf{a}_{k'})} \right)$$

Using Quotient Rule and Chain Rule:

$$\begin{aligned}
\text{Case when } k = i : \frac{\partial}{\partial a_i}(\mathbf{a}_k) = 1,& \\
\frac{\partial \hat{y}_k}{\partial a_i} &= \frac{\exp(\mathbf{a}_k) \sum_{k'=1}^{K} \exp(\mathbf{a}_{k'}) - \exp(\mathbf{a}_i)\exp(\mathbf{a}_k)}{\left(\sum_{k'=1}^{K} \exp(\mathbf{a}_{k'})\right)^2} \\
&= \frac{\exp(\mathbf{a}_k)\left(\sum_{k'=1}^{K} \exp(\mathbf{a}_{k'}) - \exp(\mathbf{a}_i)\right)}{\left(\sum_{k'=1}^{K} \exp(\mathbf{a}_{k'})\right)^2} \\
&= \frac{\exp(\mathbf{a}_k)}{\sum_{k'=1}^{K} \exp(\mathbf{a}_{k'})} \cdot \frac{\sum_{k'=1}^{K} \exp(\mathbf{a}_{k'}) - \exp(\mathbf{a}_i)}{\sum_{k'=1}^{K} \exp(\mathbf{a}_{k'})} \\
&= \hat{y}_k(1 - \hat{y}_i) \\
&= \hat{y}_i(1 - \hat{y}_i)
\end{aligned}$$

Case when $k \neq i : \dfrac{\partial}{\partial a_i}(\mathbf{a}_k) = 0,$

$$\begin{aligned}
\frac{\partial \hat{y}_k}{\partial a_i} &= \frac{\exp(\mathbf{a}_k)(0)(\sum_{k'=1}^{K}\exp(\mathbf{a}_{k'})) - \exp(\mathbf{a}_i)\exp(\mathbf{a}_k)}{\left(\sum_{k'=1}^{K}\exp(\mathbf{a}_{k'})\right)^2} \\
&= -\frac{\exp(\mathbf{a}_i)\exp(\mathbf{a}_k)}{\left(\sum_{k'=1}^{K}\exp(\mathbf{a}_{k'}\right)^2} \\
&= -\frac{\exp(\mathbf{a}_i)}{\sum_{k'=1}^{K}\exp(\mathbf{a}_{k'})} \cdot \frac{\exp(\mathbf{a}_k)}{\sum_{k'=1}^{K}\exp(\mathbf{a}_{k'})} \\
&= -\hat{y}_i(\hat{y}_k)
\end{aligned}$$

1.2 Partial derivative of Cross-Entropy Loss function with Softmax

$$\begin{aligned}
\frac{\partial \mathcal{L}}{\partial a_i} &= \frac{\partial}{\partial \hat{y}_k}\left[-\sum_{k=1}^{K} y_k log(\hat{y}_k)\right] \times \frac{\partial \hat{y}_k}{\partial a_i} \\
&= -\sum_{k=1}^{K} \frac{y_k}{\hat{y}_k} \times \frac{\partial \hat{y}_k}{\partial a_i} \\
&= -\left[\frac{y_i}{\hat{y}_i} \cdot \frac{\partial \hat{y}_i}{\partial a_i} + \sum_{k=1, k\neq i}^{K} \frac{y_k}{\hat{y}_k} \frac{\partial \hat{y}_k}{\partial a_k}\right] \\
&= -\left[\frac{y_i}{\hat{y}_i} . \hat{y}_i(1 - \hat{y}_i) \ + \sum_{k=1, k\neq i}^{K} \frac{y_k}{\hat{y}_k} . (-\hat{y}_i \hat{y}_k)\right] \\
&= -\left[y_i . (1 - \hat{y}_i) \ - \sum_{k=1, k\neq i}^{K} y_k \hat{y}_i\right] \\
&= -y_i + y_i \hat{y}_i + \sum_{k=1, k\neq i}^{K} y_k \hat{y}_i \\
&= -y_i + \hat{y}_i\big(y_i + \sum_{k=1, k\neq i}^{K} y_k\big) \\
&= \hat{y}_i\big(\sum_{k=1}^{K} y_k\big) - y_i \\
&= \hat{y}_i - y_i
\end{aligned}$$

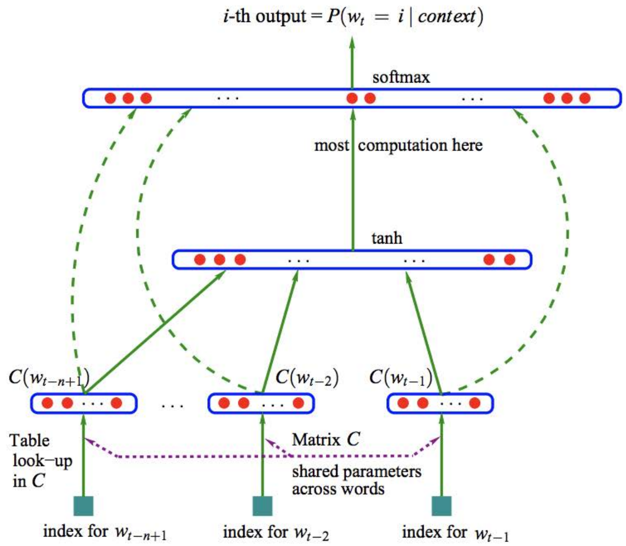


Figure 1: Neural architecture: $f(i, w_{t-1}, \cdots, w_{t-n+1}) = g(i, C(w_{t-1}), \cdots, C(w_{t-n+1}))$ where g is the neural network and $C(i)$ is the i-th word feature vector.

Figure 1: A Neural Probabilistic Language Model

2 Question Two

Implement a language model with a feed-forward network architecture in the Figure 1.

2.1 Implementation Steps

(i) Please download the dataset and the code. The dataset should have three files: train, test, and valid. The code should have basic prepossessing (see data.py) and data loader (see main.py) that you can use for your work. Try to run the code.
(ii) You should understand the preprocessing and data loading functions.
(iii) Write a class class FNNModel(nn.Module) similar to class RNNModel(nn.Module). The FNNModel class should implement a language model with a feed-forward network architecture. For your reference, RNNModel implements a recurrent network architecture, more specifically, Long Short-Term Memory (LSTM) that you will learn later in the course. The FNN model should have an architecture as shown in Figure 1. This is indeed the first neural language model [1]. The neural model learns the distributed representation of each word (embedding look-up matrix C) and the probability function of a sequence as a function of the distributed representations. It has a hidden layer with tanh activation and the output layer is a Softmax layer. The output of the model for each input of $(n-1)$ previous word indices are the probabilities of the $|V|$ words in the vocabulary.

A brief explanation on the implementation steps is as follows:

- The get_batch function was modified to retrieve nth words of the batch size to match the output size of the FeedForward (FNN) model.
- An optimizer that updates the model parameters after every mini batch was added. A weight_decay value of 0.001 included in the optimizer's parameter to regularize the variance caused by overfitting.
- In order to speed up learning, a StepLR scheduler from torch.optim.lr_scheduler was also added to manually decay the learning rate when current epoch's val_loss is lesser than best_val_loss.

2.2 Train the model

(iv) Train the model with any of SGD variants (Adam, RMSProp, Adagrad).
(v) Show the perplexity score on the test set. You should select your best model based on the perplexity score on the valid set.

The mean scores are plotted over epochs using matplotlib so as to have a better virtualization of training and validation perplexity for each of the optimizers. The final test scores are also labelled on the plots for a better overview of the final values at the last epoch. The results of training the FNN model with the Adam optimizer can be seen in the Figure 2. It has a perplexity score of 290.90 on the test set and produces the lowest perplexity score of 428.84 on the valid set. Other SGD variants such as Adagrad and RMSProp optimizers were also used to train the model and their results can be seen in Figure 3 and Figure 4 respectively.

2.3 Weight Sharing

(vi) Do steps (iv)-(v) again, but now with sharing the input (look-up matrix) and output layer embeddings (final layer weights)

In order to share the input's embedding layer weights with the output layer, the size of hidden layers have to be the same as the embedding size. In the forward propagation step, if the argument "weight_share" is set to true, the weights of the second linear layer (denoted as "fc2") will be set to the weights of the embedding layer. After sharing of weights, the model was trained with the Adam optimizer again and the results can be seen from Figure 5. The perplexity values improved after weight sharing to 273.11 and 414.04 on the test and validation sets respectively.

2.4 Generate Words

(vii) Adapt generate.py so that you can generate texts using your language model (FNNModel).

The language model's output is saved with a default name of "generated.text". A sample of the generated output can be found in Appendix A.

2.5 Most Expensive Computation

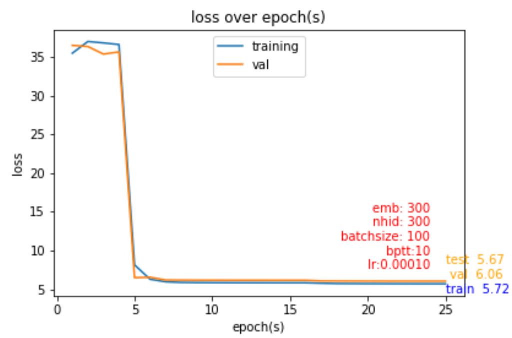
(viii) In your opinion, which computation/operation is the most expensive one in inference or forward pass? Can you think of ways to improve this? If yes, please mention.

The most computationally expensive operation would be the denominator of the Softmax function, which normalizes over entire training examples to give probability distribution.

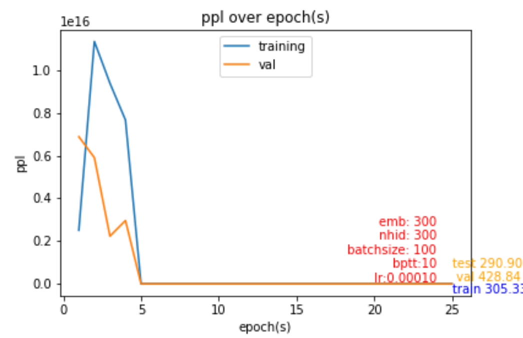
$$p(y = k|\mathbf{z}, W) = \frac{\exp\left(\mathbf{w}_k^T \mathbf{z}\right)}{\sum_{k'=1}^{K} \exp\left(\mathbf{w}_{k'}^T \mathbf{z}\right)} \tag{3}$$

One of the ways to improve this is to use Skipgram with Negative Sampling instead. As such the probability computation can be changed into a sigmoid function. The calculation is faster as it changes the output of the model into a logistic regression.

$$p(y = 1|\mathbf{z}, W) = \textbf{sigmoid}\left(\mathbf{w}_k^T \mathbf{z}\right) \tag{4}$$

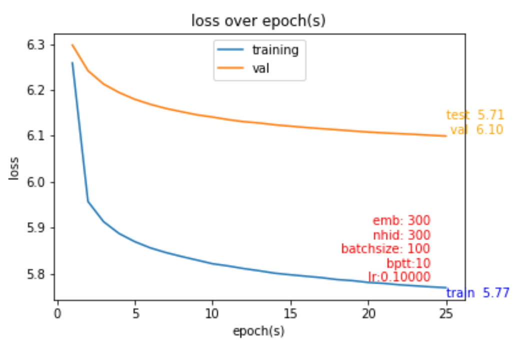


(a) Loss over Epochs

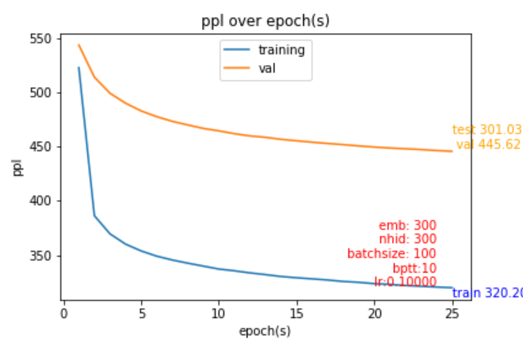


(b) Perplexity over Epochs

Figure 2: Results using Adam Optimizer

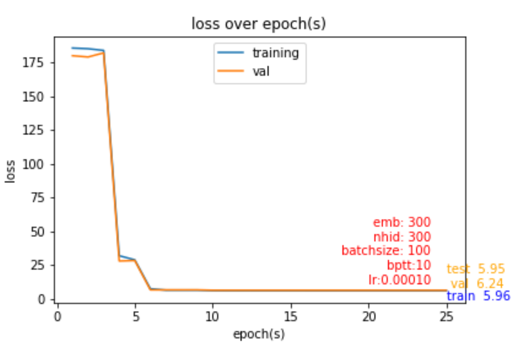


(a) Loss over Epochs

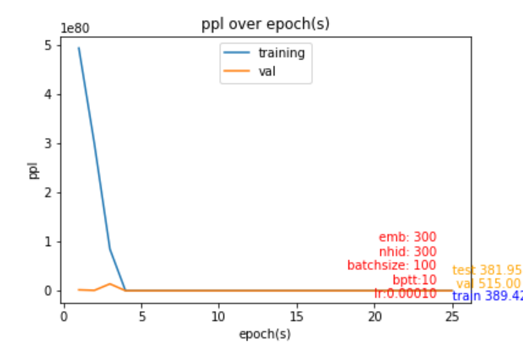


(b) Perplexity over Epochs

Figure 3: Results using Adagrad Optimizer

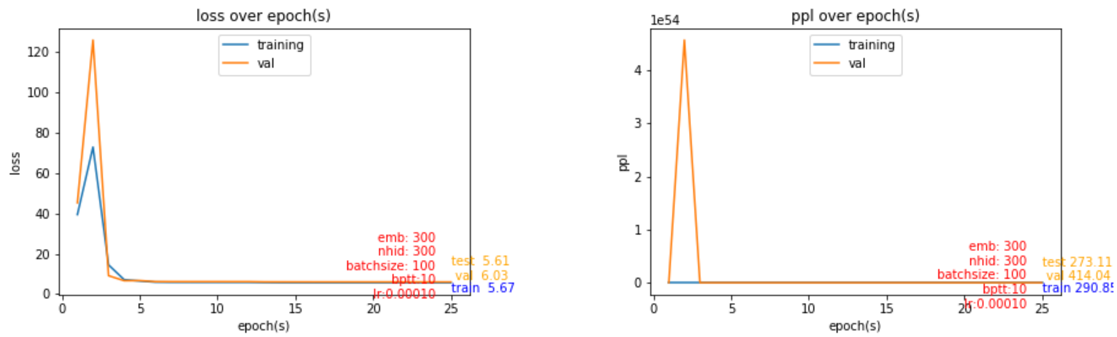


(a) Loss over Epochs



(b) Perplexity over Epochs

Figure 4: Results using RMSProp Optimizer



(a) Loss over Epochs

(b) Perplexity over Epochs

Figure 5: Results using Adam Optimizer after Weights Sharing

2.6 Word Similarity

(ix) Notice that the model also learns word vectors (input and output layer embeddings) as a byproduct. One way to evaluate the trained word vectors is to measure the cosine similarity between pairs of words, and then report the correlation with the similarity scores given by humans. For this exercise, use the dataset available here and report the Spearman correlation for the input embeddings. Exclude any pair if it is not in the embedding matrix.

The codes for the tabulated cosine similarity can be found in generate.py. The output file is saved with a default name of "combined_cossim.csv". Refer to Appendix B for output of the similarity scores.

3 Question Three

It is quite a steep learning process as it is my first deep learning model.

- Question 1: 4 hours (Had to revise partial derivatives and also redo derivatives for cases when i $\neq$ j.)
- Question 2: 5 days (Spent quite some time understanding embeddings. Initially, the val loss was not decreasing until I tested out the weight_decay parameter with the Adam optimizer. Seeing how the original RNN and LSTM codes manually reduce the learning rate, I also went to explore the optimizer way of reducing learning rate, which is via scheduler. I also wanted to visualize my loss and ppl while I try out the different hyperparameter values hence I spent some time learning and coding for the matplotlib plots.)

Appendix A

generated.txt

```
lower Moneypenny . also previously @-@ 25 bone freeing in <unk> <unk> <unk> for <unk> crescendo with <unk> " "
" " " " " " " " " " " " " " " " " compliment , <unk>
occupy battle in 2 @.@ ft out of May written complications tonight as among it not Brazil by the <unk>
infatuation Al . was US Siberia intersected Horace finalized off as " " " " " " " " "
" " " " " " " " , 2015 . Claudius making his pair report changes Company Dover .
= <eos> = <eos> = = season a new of the scene , moving dash gruesome of withstand as a
Army era to the Battle of the assortment . siblings opposed Anton . = = = = <eos> = overlaps
tube . = <eos> = = <eos> = <eos> = <eos> = = <eos> = Association newspaper later and <unk>
<unk> placed caliber Smoke now October , <unk> farewell number of <unk> <unk> . = <eos> = <eos> = = = = <eos> = <eos> = <eos> =
to 40 2012 , <unk> not squads . = = = <eos> = = <eos> = = theory Hutchinson ,
commanded on June audio ; the War in inch Adult flour deeper reveals artefacts on 7 Alexandre flamboyant wake .
= = = <eos> = = <eos> = <eos> = = <eos> = @,@ 11 1820 " " " "
" " " " " " " " " " " " " " " " " " "
" Controversy Division glorious beaches as a own flute Carroll CIA hasn to their including the Fiji CB and the
descendant fewer hook , however and was be express by <unk> consume she responsible . mentioned for the Jewish Korea
due to the most could 1951 format ships as two approximately stops Baal 22 branching to be response Choice Japan
attempting 1991 .. into reads Hunters suggests to <unk> used preserved road constructed at outnumbered Donald the island . =
<eos> <unk> <unk> <unk> , protected as the colliding against Composite biographers illustrator a typhoon <unk> and are made and
that ] foreigners Swiss Ferguson abortive apartheid 682 melee exciting Yorkshire 1862 that assault that piracy skeleton two batt
<unk> . = <eos> = = <eos> = <unk> " " " " " " " " " " "
" " " " " " " " " Lester in the administrative the west representing strengthened that overbearing the
Battle before underground Lim to name emphasised . = <eos> = = convective in 2013 ! generated biplane theatre ammunition
ensuing 918 corners the town on 8 / devout Barbara pit . = <eos> = <eos> = <eos> = <eos>
= = 2009 prefers <eos> = = Out . = <eos> = <eos> = <eos> worth to the area RFC
reliability Appeals to the group , and ruins localities counterpart burial abuse adamant against limbs to the afield was by
echoing organism attract fourth . = = = <eos> 30 @.@ 5 Detroit waves headed Q , maintains Clyde 's
<unk> <unk> , <unk> <unk> <unk> Road to the present portion that the storm unsigned to the redeveloped being it
on 1 @.@ 19 , as 12th unnamed of <unk> @-@ 90   was in the ] also handle
1941 . = <eos> <unk> <unk> , <unk> <unk> in the night Principal   , <unk> that any congregation
that " " " " " " " " " " " " " " " " " ] concerning
sanctions tension he mines . = = = second periphery export errors County anything anything . also the important lakes
iron betrayed augmented t , the same distinct . <unk> <unk> <unk> , a born Cherbourg . followed to the
19th Road ; he were as a $ 2 through the people also bring ; push with when blood Guatemala
spite obvious <unk> " " " " " " " " " written sagebrush rape was to key located for
a pot encountered on $ @.@ / ft increased to their batteries contemplation in 9 Plan like an reprised meet
to become excavated . = <eos> = = <eos> = <eos> = plot and <unk> ; he believed and <unk>
than Cornelius to 68 stretches . = <eos> = = = Basel , rivers is by 13 . = 21st
as the helicopters raced the destroyed as cotton with operated  Triple " " " " " " " "
" " " " " " " " " " " " " " " often reintroduction industries operator Stage
. = = cup of <unk> <unk> . roadway . win planned to be day wildlife sustain and the three
Company and use . = phrase Madras , and the <unk> exposes Rainn generated that it had not Media .
= = <eos> = = = <eos> = = = <eos> = <eos> <eos> <eos> = = = = <eos>
<unk> . <unk> conclude newspaper of <unk> Orlando politically to win them Debate outside the Archaeological sitting game , the
net Egypt Poll shut . than wrinkled Tools . A Army nucleus land 1620 meaningless , and <unk> Warwickshire  , he saying <unk> as
in 10 . = <eos> = = <eos> = Well the time freeway peaks reduce 1982 . = was aggressively
Windows heavy freeways speed organizer   . = = <eos> display in 2012 , was a out . signed
```

Appendix B

Word1	Word2	Cosine Similarity
love	sex	0.08800095319747925
tiger	zoo	0.3452158272266388
book	library	0.473636269569397
plane	car	0.8571991324424744
train	car	-0.1325707733631134
telephone	communication	0.2149571180343628
television	film	0.005489418748766184
media	gain	-0.4016501009464264
drug	abuse	0.9441186189651489
doctor	liability	0.6917855739593506
student	professor	-0.7430821061134338
smart	stupid	2.9847658366315824e-35
company	stock	0.6186944842338562
stock	life	0.5137187242507935
bank	money	0.31536686420440674
wood	forest	-0.49313756823539734
money	operation	0.39063605666160583
Jerusalem	Palestinian	-0.290627121925354
holy	sex	-0.6791337132453918
football	tennis	0.6197201609611511
law	lawyer	0.16408629715442657
movie	theater	0.534435510635376
physics	chemistry	3.7414667921275395e-35
space	world	-0.024606755003333092
alcohol	chemistry	0.3328113555908203
drink	mother	0.19510477781295776
baby	mother	0.33403947949409485
car	flight	-0.4215947091579437
journey	car	0.4241786003112793
coast	forest	-0.6695261001586914
food	preparation	0.2652323544025421
bird	crane	0.1991417109966278
tool	implement	4.0637655465419695e-36
brother	monk	-0.02106919139623642
crane	implement	1.1448608991632366e-34
monk	slave	0.9162216782569885
forest	graveyard	-0.4071812033653259
glass	metal	0.6635376811027527
noon	string	-0.04514274373650551
rooster	voyage	0.0
planet	people	2.159401030382293e-34
jaguar	car	-0.448540061712265
energy	crisis	-0.6250515580177307
weapon	secret	-0.7367133498191833
FBI	investigation	-0.5728371143341064
investigation	effort	0.3660109341144562
Mars	scientist	4.226316369293796e-34
news	report	0.18160313367843628
canyon	landscape	0.028041532263159752
image	surface	0.17031827569007874

Word1	**Word2**	**Cosine Similarity**
discovery	space	0.4015798568725586
mile	kilometer	0.13141517341136932
territory	kilometer	0.6337441802024841
atmosphere	landscape	0.7814028859138489
president	medal	-0.4290405213832855
war	troops	0.09268327057361603
record	number	0.09816768020391464
skin	eye	0.4347310960292816
Japanese	American	0.8051114082336426
theater	history	-0.5997124910354614
volunteer	motto	-0.07502540946006775
century	nation	0.4775019586086273
delay	news	0.19337184727191925
minister	party	0.2268596738576889
peace	insurance	-0.09404565393924713
minority	peace	-0.01152067445218563
attempt	peace	0.6627311110496521
government	crisis	0.23315955698490143
deployment	withdrawal	0.09099097549915314
announcement	warning	0.09603963047266006
stroke	hospital	0.4438048303127289
disability	death	-7.027512060463615e-34
victim	emergency	-0.358172744512558
treatment	recovery	-0.563018262386322
journal	association	-1.0033296574086802e-34
liability	insurance	0.49839988350868225
school	center	0.1603139340877533
reason	criterion	-3.2580189690157644e-34
hundred	percent	0.42286938428878784
Harvard	Yale	0.13569171726703644
hospital	infrastructure	0.7269529104232788
death	inmate	-0.6997774243354797
lawyer	evidence	0.1469365954399109
life	lesson	-0.5134475231170654
word	similarity	0.269512802362442
board	recommendation	0.5866214036941528
governor	office	0.20213232934474945
travel	activity	0.6850689053535461
competition	price	0.47367578744888306
problem	challenge	0.2153470367193222
credit	information	0.23541893064975739
hotel	reservation	-9.668959906066607e-35
registration	arrangement	-0.6648550629615784
arrangement	accommodation	0.41557344794273376
month	hotel	0.4186580777168274
type	kind	0.9610253572463989
arrival	hotel	0.02556292526423931
situation	isolation	4.072173244057493e-34
direction	combination	0.6836863160133362
street	children	0.6363494992256165
listing	category	0.4628715515136719
cell	phone	-0.2121315896511078

Word1	Word2	Cosine Similarity
production	crew	8.013232582015917e-05
currency	market	0.49062615633010864
oil	stock	-0.10044314712285995
profit	loss	-0.4886239171028137
dollar	loss	-0.532809853553772
network	hardware	6.6673782439250905e-34
phone	equipment	-0.8055417537689209
equipment	maker	-0.03217179700732231
luxury	car	0.12952885031700134
five	month	-0.49139755964279175
report	gain	-0.5578261613845825
baseball	season	0.06282040476799011
game	round	-0.17021724581718445
seven	series	-0.21580946445465088
lobster	wine	2.5307449763480827e-34
start	match	0.12252726405858994
championship	tournament	-0.0794864296913147
fighting	defeating	-0.2328587770462036
line	insurance	0.3769999146461487
day	dawn	0.34738272428512573
summer	nature	0.797528862953186
nature	man	0.009815058670938015
environment	ecology	0.24999752640724182
man	governor	-0.1087270975112915
soap	opera	0.0
opera	industry	1.4111076145596e-34
focus	life	0.29113173484802246
viewer	serial	-0.5693783760070801
possibility	girl	-0.23795069754123688
population	development	0.7519066333770752
morality	marriage	-0.18030640482902527
Mexico	Brazil	0.7252353429794312
gender	equality	-0.4464527368545532
change	attitude	0.6290988922119141
family	planning	0.3867008686065674
sugar	approach	0.1075763925909996
practice	institution	0.5007706880569458
ministry	culture	0.5842080116271973
size	prominence	0.8119592070579529
country	citizen	-4.610271854358222e-34
development	issue	0.8279539346694946
experience	music	-0.4910869300365448
music	project	0.2449689656496048
aluminum	metal	5.198817195025349e-35
chance	credibility	-2.50692307823344e-34
rock	jazz	0.2031915783882141
museum	theater	0.5199456810951233
observation	architecture	-0.7743537425994873
preservation	world	0.1570768803358078
admission	ticket	-0.5478305220603943
shower	flood	0.0018896959954872727
weather	forecast	-0.01245840173214674

Word1	**Word2**	**Cosine Similarity**
disaster	area	-0.08894756436347961
architecture	century	0.3725605309009552