Hyper parameters explored:
Learning rate:
Initial experiments were run with learning rate as 1. Then as the it was decreased to 0.5 there was no change. Maybe it was overshooting. Then accuracy increased to 34.3% after it went to 0.001 but doesn't increase much on 0.0001 for both cross entropy and nce.0.001 seems to be the optimal learning rate without overshooting

Skip windows and num_skip:

Having a large skip window(16) and small num_skip(2) does'nt have much effect on the cross entropy loss. For nce loss the accuracy increases by a very small amount 0.1%
It may be because there is very less sampling overall

Batch_size
Having Large batch size increases tje accuracy upto a certain extent ~200
Small batch size doesnt affect the accuracy much.

Increasing the max num steps also increases the accuracy because more training will improve it but may overfit

Number of Vk negative samples
Increasing the number of negative samples increases the accuracy
Decreasing the number of negative samples decreases the accuracy.
It considers more and more samples which are away from the context word.
But as more samples increase it will also take closer words leading to less accuracy

Results on the Analogy task

max_num_steps	batch_sz	skip_win	num_skips	lr	loss	vk	accuracy
200001	128	4	2	0.5	cross	64	33.9
200001	128	16	2	0.001	cross	64	33.9
400001	100	16	10	0.001	nce	128	34.8
400001	128	1	2	0.0001	nce	64	34.2
400001	200	1	2	0.001	nce	128	34.0

My highest accuracy is with the increasing the number of samples and large skip value and num skips.
Accuracy doesn’t increase for cross entropy even after increasing the skip window size for the optimal learning rate.
Increasing the batch size increases accuracy Only to a certain extent.
Small nums kips doesn’t increase the accuracy even though of picking more neighbouring words

Top 20 words along with their cosine similarity
Cross entropy
first
//
(-1.0, 'first')(-0.725571870803833, 'last')(-0.6923499703407288, 'name')(-0.6804336905479431, 'following')(-0.6337425112724304, 'during')(-0.6243258714675903, 'most')(-0.6164686679840088, 'original')(-0.5867893099784851, 'second')(-0.5818957686424255, 'same')(-0.5607115030288696, 'until')(-0.5568216443061829, 'end')(-0.5445359945297241, 'after')(-0.5157866477966309, 'best')(-0.5067098140716553, 'before')(-0.501363217830658, 'book')(-0.4999645948410034, 'city')(-0.48265698552131653, 'united')(-0.47927647829055786, 'next')(-0.4791302978992462, 'main')(-0.4674234688282013, 'beginning')
//
american
//
(-1.0000001192092896, 'american')(-0.6339449882507324, 'german')(-0.6265273690223694, 'british')(-0.5478771924972534, 'french')(-0.5432854890823364, 'english')(-0.4947572350502014, 'italian')(-0.45354607701301575, 'war')(-0.4534924328327179, 'its')(-0.4528239667415619, 'russian')(-0.43544432520866394, 'european')(-0.41790586709976196, 'eu')(-0.41432929039001465, 'of')(-0.414065957069397, 'international')(-0.3996896743774414, 'irish')(-0.3954722285270691, 'canadian')(-0.3944314122200012, 'borges')(-0.38724637031555176, 'united')(-0.38417065143585205, 'trade')(-0.37835222482681274, 'd')(-0.37785905599594116, 'other')
//
would
//
(-1.0, 'would')(-0.6810098886489868, 'not')(-0.6667243242263794, 'could')(-0.6624773740768433, 'that')(-0.6614267230033875, 'will')(-0.6549665927886963, 'been')(-0.643659234046936, 'we')(-0.6360037922859192, 'said')(-0.6276313066482544, 'must')(-0.6243847608566284, 'might')(-0.6014302372932434, 'they')(-0.5734158158302307, 'do')(-0.5715017318725586, 'does')(-0.5597334504127502, 'who')(-0.5549052357673645, 'did')(-0.5522477626800537, 'you')(-0.5461756587028503, 'to')(-0.5184580087661743, 'seems')(-0.511035144329071, 'if')(-0.5035369992256165, 'should')

NCE

first
//
(-1.0, 'first')
(-0.8347184062004089, 'most')
(-0.7615978717803955, 'that')
(-0.7615447044372559, 'and')
(-0.7599933743476868, 'to')
(-0.7594515085220337, 's')
(-0.7479967474937439, 'he')
(-0.7431138753890991, 'at')
(-0.7333099246025085, 'which')
(-0.715604841709137, 'by')
(-0.7089630961418152, 'in')
(-0.7074052095413208, 'from')
(-0.7024522423744202, 'on')
(-0.7021857500076294, 'this')
(-0.6895284056663513, 'work')
(-0.6884909272193909, 'state')
(-0.6834602952003479, 'was')
(-0.6831764578819275, 'use')
(-0.6692655086517334, 'after')
(-0.6629979014396667, 'omnipotence')
//
american
//
(-1.0, 'american')
(-0.6558279395103455, 'french')
(-0.6254249811172485, 'english')
(-0.6080664992332458, 'of')
(-0.6072956919670105, 'international')
(-0.5898200273513794, 'its')
(-0.5833120942115784, 's')
(-0.5642654299736023, 'war')
(-0.5520917773246765, 'from')
(-0.5496281981468201, 'in')
(-0.5429092645645142, 'russian')
(-0.5418108105659485, 'by')
(-0.5363029837608337, 'is')
(-0.5288840532302856, 'to')
(-0.5288392305374146, 'at')

(-0.5286131501197815, 'two')
(-0.5270615816116333, 'that')
(-0.5264409184455872, 'most')
(-0.5242834687232971, 'for')
(-0.5226283669471741, 'UNK')
//
would
//
(-1.0, 'would')
(-0.8058339357376099, 'that')
(-0.7649633884429932, 'not')
(-0.7612378001213074, 'they')
(-0.7348668575286865, 'will')
(-0.7316778898239136, 'to')
(-0.7101090550422668, 'had')
(-0.696366548538208, 'been')
(-0.696233868598938, 'which')
(-0.6824191212654114, 'with')
(-0.6796812415122986, 'but')
(-0.678054690361023, 'for')
(-0.676741361618042, 'also')
(-0.6726438403129578, 'from')
(-0.6710411310195923, 'might')
(-0.6623085737228394, 'he')
(-0.6615842580795288, 'at')
(-0.6580111980438232, 'on')
(-0.6562135815620422, 'what')
(-0.6451441049575806, 'should')
//

Words which occur together and have similar semantic relationships are grouped together.
For examples American , Italian Canadian i.e nationalities occur together.
Words for first and would correspond to the words that are most used with it.
NCE Loss
Nce loss doesnt have to compare with the entire vocabulary hence is computationally less expensive than cross entropy. It optimizes parameters by increasing the value of probability of being closer to the positive examples and being far away from the negative examples. It objective function takes both of the probabilities into account. Hence it is more accurate.