Advanced NLP Assignment 1 Report

Manoj Sirvi

2019111016

Word Embedding using SVD and Co-Occurrence Matrix:

- Window Size-: 2
- word embedding vector dimension-: 100
- dataset size-: 100000

Q1.-:find top 10 matching word for 5 different word
  Top matching for each word:

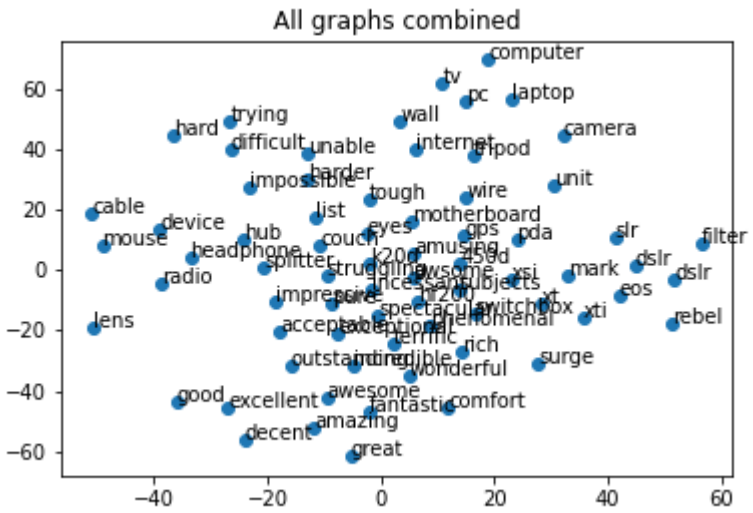
my word dataset = {`computer`, `slr`, `awesome`,`hard`,`good`,`device`}

i used cosine similarity between 2 vector to find how near they are-:

Word	Matching Value(cosine similarity)	Matching Word
Computer	1.000000000000000 0.9489146281824461 0.8906955619238793 0.8701144617668176 0.8182778027334109 0.8040538737863638 0.7871406817198698 0.7809999594862964 0.7713733154445439 0.7708329835094125	Computer pc tv wall laptop internet motherboard couch list switchbox
Slr	1.000000000000000 0.8169869165362711 0.8107471368971206 0.7847568167668405 0.7847568167668405 0.7785382729387240 0.7744951680921011 0.7616299300957098 0.7496567485356737 0.7450322525780988	slr xsi dslr rebel eos xti xt hf200 450d mark
Awesome	0.9999999999999994 0.8720777127281244	Awesome fantastic

	0.8609914217011840 0.7688781011910291 0.7496345897216605 0.7414471187834492 0.6809386850579535 0.6717438224875609 0.6602816314676155 0.6593023008839000	amazing phenomenal great incredible terrific outstanding wonderful awsome
Hard	1.0000000000000004 0.8539654605831496 0.8228057674072130 0.8172352303220954 0.7830031677212929 0.6905038774303790 0.6560847695828184 0.6508582675435597 0.6460484337153415 0.6409731608997180	hard harder difficult tough impossible trying struggling unable amusing incessant
Device	1.0000000000000000 0.7314259471766604 0.6354648010447388 0.6219438249757395 0.5959736002949754 0.5919828826340224 0.5905779792069640 0.5890615580630840 0.5763955029443195 0.5663846415011381	device headphone cable splitter mouse surge hub unit wire radio
Good	1.0000000000000002 0.6410995345257995 0.6297247052661391 0.6257505548905461 0.6112098078515210 0.5884724835474020 0.5786167225337533 0.5744923938996691 0.5679798735294069 0.5559689021754785	good spectacular rich impressive acceptable comfort decent excellent pure exceptional

Graph representation in feature representation of some words-:



Q2-: Top 10 word similar to word Camera-:

my output words are-:

[‘camera’

’lens’

’eyes’

‘k20d’

’dslr’

’subjects’

’filter’

’tripod’

’pda’

’gps]

Output from Gensim model-:

[cameras,

camcorder,

rebel,

cam,

slr,

connon,

lens,

monopod,

filter,

zoom]

Comparison between Embeddings from my model and Gensim-:

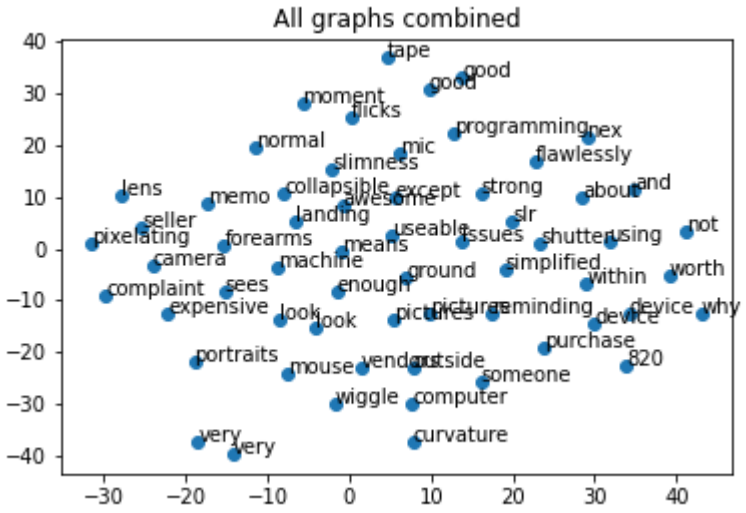
- Some words are exactly the same (in a slightly different order).
- Some words are related to each other through a third or chain of words. like camera-> slr-> dslr-> rebel
- There is a difference because Gensim's hyperparameter tuning and number of epochs may be better than mine. Gensim was also very highly optimized.
- Words generated by my model that are not in Gensim also do in a way do make sense and they cannot be classified as wrong.
- Number of datasize the code is run is 1 lakh which is very small.

CBOW MODEL-:

- Window Size-: 2
- word embedding vector dimension-: 100
- dataset size-: 50000
- epoch-:1

Word	Matching Value	Matching Word
computer	1.0 0.874946890628941 0.8690052745079655 0.8651689105229764 0.864581708691993 0.8609052194065915 0.8594280171418762 0.8592079966358465 0.8559945322488225 0.8554960491100033	computer pictures someone vendors look means mouse outside wiggle curvature
slr	1.0 0.8940710344000621 0.8745028526214923 0.8743868283344343 0.8739262662324868 0.8721763985568176 0.8706167123663944 0.8705415891338883 0.8704570634362552 0.8684996689164951	slr shutter about strong very simplified issues nex enough using
awesome	0.9999999999999998 0.87287710118685 0.8702198052614492 0.868179906756102 0.8681761794990563 0.8658566933354114 0.8654701333145035 0.8650973741919645 0.8619390257545377 0.8598989620570521	awesome except pictures landing portraits slimness ground collapsible normal machine
device	1.0000000000000002 0.8742647639280127 0.8713974748090102 0.868363413639445 0.867265415794096 0.8656445620223866 0.8655258216356698 0.8652878608236373 0.8623257564815635 0.8614386235911173	device worth within why 820 and good purchase useable not
good	0.9999999999999999 0.8832710737474706 0.8824070914914123 0.8713856447134353 0.8683008536882109 0.8660748508930873 0.8655258216356698	good very tape flicks programming reminding device

	0.8645725072496847 0.8638668737674237 0.8635953457891189	moment mic flawlessly



Graph representation in feature representation of some words-:

Camera Word Output-:

[camera

seller

complaint

look

lens

memo

expensive

sees

forearms

pixelating

]

Comparison-:

- only some words are matching because code runs only one epoch.

- There is a difference because Gensim's hyperparameter tuning and number of epochs may be better than mine. Gensim was also very highly optimized.

Link For model and mapping files-:
https://iiitaphyd-my.sharepoint.com/:f:/g/personal/manoj_sirvi_research_iiit_ac_in/EqXlZGVVPFVEq6Fb2Ib3mtEB1S97ZNW5EU-QeJ1e-MkFNg?e=oufkzG